

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Testovací sonda pro automatický monitoring přenosových
parametrů v mobilní síti**

**Test Probe for Automatic Monitoring of Transmission
Parameters in Mobile Network**

2015

Bc. Lukáš Kubný

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student:

Bc. Lukáš Kubný

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T059 Mobilní technologie

Téma:

Testovací sonda pro automatický monitoring přenosových parametrů v
mobilní síti
Test Probe for Automatic Monitoring of Transmission Parameters in
Mobile Network

Zásady pro vypracování:

1. Proveďte rešerši faktorů ovlivňující kvalitu služby pro mobilní technologii GSM, UMTS a LTE.
2. Navrhněte a naprogramujte mobilní aplikaci pro OS Android, která bude automaticky měřit aktuální hodnotu dosaženého zpoždění a přenosové rychlosti v mobilní síti. Výsledky se budou automaticky zasílat na vybraný server.
3. Navrhněte funkci watchdog na hardwarové a softwarové úrovni pro vyvinutou aplikaci.
4. Vytvořenou aplikaci odlaďte a otestujte v reálné síti GSM, UMTS a LTE.

Seznam doporučené odborné literatury:

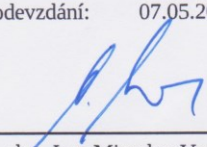
UJBÁNYAI, Miroslav. Programujeme pro Android. Vyd. 1. Praha: Grada, 2012, 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.

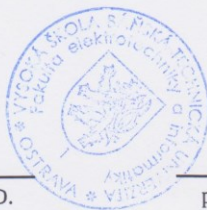
Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015


doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

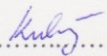



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 29. dubna 2015

..........
podpis studenta

Poděkování

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Liborovi Michalkovi, Ph.D. za odbornou pomoc, rady a konzultace při vytváření této diplomové práce.

Abstrakt

Tato diplomová práce pojednává o testovací sondě pro automatický monitoring přenosových parametrů v mobilní síti. Toto téma jsem si zvolil, realizace je součástí řešení.

V úvodu diplomové práce se zabývám teoretickým rozбором tématu, především provádím rozbor faktorů, které ovlivňují kvalitu služby pro mobilní technologii GSM, UMTS a LTE. Další část diplomové práce je tvořena návrhem a realizací aplikace pro operační systém Android, která má za úkol automaticky měřit aktuální hodnotu dosaženého zpoždění a přenosové rychlosti v mobilní síti. Výsledky svých měření aplikace automaticky odesílá na vzdálený server. Součástí řešení je také funkce "watchdog" na hardwarové a softwarové úrovni a otestování aplikace v reálné síti GSM, UMTS a LTE včetně poznatků, které jsem během řešení zjistil. Výsledkem práce je testovací sonda realizovaná pomocí běžného mobilního telefonu a platformy Arduino, která je použitelná v praxi a je mnohonásobně levnější, než komerční produkty.

Klíčová slova

GSM; UMTS; LTE; Android; Arduino; zpoždění; přenosová rychlost

Abstract

This thesis deals with test probe for automatic monitoring of transmission parameters in mobile network. This topic I have chosen, implementation is the part of the solution.

At the beginning of the thesis I deal with theoretical analysis of theme, especially I am doing analysis of factors which affect quality of service for mobile technology GSM, UMTS, LTE. Next part of the thesis consists of draft and solution of application for operation system Android, which aims to automatically measure the current value reached by the delay and transmission speed in the mobile network. The results are automatically sent to a remote server. The solution also features a watchdog on hardware and software level and testing of application in the real GSM, UMTS and LTE networks including knowledge which I found out during the solution of it.

The result of this thesis is the test probe realised by an ordinary mobile phone and the Arduino platform, which is applicable in practice and much cheaper than commercial products.

Key words

GSM; UMTS; LTE; Android; Arduino; delay; transmission speed

Seznam použitých zkratk

Zkratka	Význam	Český ekvivalent
AT	Attention Commands	AT příkazy
BER	Bit Error Rate	Bitová chybovost
BSC	Base Station Controller	Ovladač základnových stanic
BTS	Base Transceiver Station	Vysílací základnová stanice
CEPT	European Conference of Postal and Telecommunications Administrations	Evropská konference poštovní a telekomunikační správy
CSD	Circuit Switched Data	Data s přepojováním okruhů
DECT	Digital Enhanced Cordless Telecommunications	Digitální bezšňůrová telekomunikace
DVB-T	Digital Video Broadcasting - Terrestrial	Digitální obrazové pozemní vysílání
DVM	Dalvik Virtual Machine	Virtuální stroj Dalvik
EDGE	Enhanced Data Rates for GSM Evolution	Rozšířené datové sazby pro GSM evoluci
eNodeB	Evolved NodeB	Vyvinutá základnová stanice
ETSI	European Telecommunications Standards Institute	Evropský institut pro telekomunikační standardy
FDMA	Frequency Division Multiple Access	Mnohonásobný přístup s frekvenčním dělením
GMSK	Gaussian Minimum Shift Keying	Gaussovské klíčování minimálního posunu
GPRS	General Packet Radio Service	Obecná rádiová služba pro přenos paketů
GPS	Global Positioning System	Systém globální navigace
GSM	Groupe Spécial Mobile; Global System for Mobile Communications	Systém pro globální telekomunikační služby
HLR	Home Location Register	Domácí lokační registr

ICMP	Internet Control Message Protocol	Protokol řídicích zpráv Internetu
IDE	Integrated Development Environment	Integrované vývojové prostředí
ISP	Internet Service Provider	Poskytovatel služeb Internetu
ITU	International Telecommunication Union	Mezinárodní telekomunikační unie
LAC	Location Area Code	Místní kód oblasti
LTE	3GPP Long Term Evolution	3GPP dlouhodobý vývoj
MOS	Mean Opinion Score	Ohodnocení na základě názoru
MS	Mobile Station	Mobilní zařízení
MSC	Mobile Switching Centre	Mobilní telefonní ústředna
OFDM	Orthogonal Frequency Division Multiplexing	Ortogonální frekvenčně dělené multiplexování
OS	Operation System	Operační systém
OSS	Operation Subsystem	Operační subsystém
PING	Packet Internet Groper	Testování dostupnosti pomocí ICMP výzvy
QOE	Quality of Experience	Kvalita zážitku
QOS	Quality of Services	Kvalita služby
QPSK	Quadrature Phase Shift Keying	Kvadraturní klíčování fázovým posunem
RIL	Radio Interface Layer	Vrstva rádiového rozhraní
RNC	Radio Network Controller	Řídicí jednotka rádiové sítě
SC-FDMA	Single Carrier - Frequency Division Multiple Access	Jedna nosná – mnohonásobný přístup s frekvenčním dělením
SMS	Short Message Service	Služba krátkých zpráv
TEMS	Test Mobile Systems	Testovač mobilních systémů
UE	User Equipment	Uživatelské zařízení
UHF	Ultra High Frequency	Ultra krátké vlny

UMTS	The Universal Mobile Telecommunications System	Univerzální mobilní telekomunikační systém
USB	Universal Serial Bus	Univerzální sériová sběrnice
UTRAN	UMTS Terrestrial Radio Access Network	UMTS pozemní rádiová síť
VoIP	Voice Over Internet Protocol	Přenos hlasu přes Internetový protokol
W-CDMA	Wideband Code Division Multiple Access	Širokopásmový vícenásobný přístup pomocí dělení kódem
WiFi	Wireless Fidelity	Typ bezdrátové sítě
WWAN	Wide Wireless Area Network	Rozsáhlá oblast s pokrytím bezdrátové sítě

Seznam použitých termínů

Termín	Význam termínu
Frekvence (kmitočet)	Udává počet opakování periodického děje za daný časový úsek. Například počet kmitů proudu či napětí za jednotku času.

Obsah

Úvod.....	- 13 -
1 Mobilní sítě	- 14 -
1.1 Rozdělení mobilních sítí dle způsobu šíření signálu	- 14 -
1.1.1 Celulární (buňkové) rádiové sítě	- 14 -
1.1.2 Sítě satelitních telefonů	- 15 -
1.1.3 Sítě bezšňůrových telefonních systémů.....	- 15 -
1.2 Rozdělení mobilních sítí dle generace vývoje.....	- 15 -
1.3 Sítě GSM, UMTS, LTE.....	- 16 -
1.3.1 GSM	- 16 -
1.3.2 UMTS.....	- 17 -
1.3.3 LTE/EPC	- 17 -
2 Kvalita služby pro mobilní technologie	- 18 -
2.1.1 Nástroje k měření přenosových parametrů.....	- 19 -
2.1.2 Parametry určující kvalitu služby	- 21 -
3 Testování a optimalizace mobilních sítí	- 24 -
4 Použité technologie	- 25 -
4.1 Android	- 25 -
4.2 Arduino	- 26 -
4.3 Mobilní telefon Sony Xperia V	- 27 -
5 Vlastní řešení.....	- 28 -
5.1 Měřicí aplikace – úvod, skript init.....	- 29 -
5.2 Měřicí skript test.sh	- 31 -
5.2.1 Přepínání typů mobilních datových sítí	- 31 -
5.2.2 Výpočet přenosové rychlosti	- 35 -
5.2.3 Způsob měření podle technické zprávy ETSI TR 102 678	- 37 -
5.2.4 Stanovení délky zpoždění (latence).....	- 38 -
5.2.5 Zjištění dalších parametrů měření	- 38 -
5.2.6 Odeslání výsledků	- 41 -
5.3 Doplnkový skript laccellid.sh.....	- 41 -

5.4	Funkce watchdog.....	- 46 -
5.4.1	Úprava operačního systému Android v měřicí sondě.....	- 47 -
5.4.2	Konstrukce hlídacího obvodu pomocí platformy Arduino.....	- 49 -
6	Testování sondy.....	- 56 -
6.1	Schéma testování.....	- 56 -
6.2	Zjištěné údaje	- 57 -
7	Naměřené výsledky	- 58 -
7.1	Výsledky měření rychlosti stahování dat sítě 2G	- 58 -
7.2	Výsledky měření rychlosti stahování dat sítě 3G	- 59 -
7.3	Výsledky měření rychlosti stahování dat sítě LTE.....	- 59 -
Závěr		- 61 -
Použitá literatura		- 63 -

Úvod

Několik posledních desetiletí je ve znamení neustálého rozvoje, vylepšování a rozšiřování telekomunikačních technologií. Výjimkou nejsou ani mobilní sítě, které si prošly postupným vývojem a lze je tak rozčlenit do několika generací podle použitých technologií a dalších parametrů.

Zadání diplomové práce bylo vytvořeno společností T-Mobile Czech Republic a.s. Tato diplomová práce se zabývá automatickým monitoringem přenosových parametrů mobilních sítí. Jedná se o parametry, pomocí kterých můžeme určit kvalitu služby. Kvalita služby, anglicky QoS (Quality of Services), je v mobilních sítích určena několika parametry, mezi nimiž je například rychlost stahování a odesílání dat či odezva.

Úkolem této diplomové práce je pak nejprve provést teoretickou rešerši faktorů ovlivňujících kvalitu služby pro mobilní technologie GSM, UMTS a LTE. Součástí bude taktéž vysvětlení, proč se vůbec zabývat testováním mobilní sítě a jaké nástroje jsou k tomuto účelu na trhu k dispozici.

Hlavní náplní diplomové práce je vytvořit testovací sondu. Tato testovací sonda má být realizována pomocí mobilního telefonu s operačním systémem Android. Má být plně samostatná a provádět veškerá měření automaticky bez jakékoliv nutnosti interakce s uživatelem. Aby mohla být testovací sonda plně automatická a samostatná, musí být součástí řešení také funkce watchdog, která bude hlídat mobilní telefon a v případě problému jej musí restartovat, aby měření mohla automaticky a bez jakéhokoliv zásahu nějaké osoby pokračovat. Z praxe jsou totiž známy případy, kdy se mobilní telefon začne časem chovat nestandardně, či může úplně „zamrznout“. Vytvořenou aplikaci je pak nutné odladit a otestovat v reálné síti GSM, UMTS a LTE.

Měření by mělo odpovídat doporučením uvedeným ve specifikaci ETSI TR 102 678. Podle této specifikace je měření realizováno nikoliv pomocí přenosu souborů s fixní velikostí, nýbrž pomocí fixně dané doby přenosu, po kterou bude probíhat stahování a odesílání dat. Po uplynutí této fixní doby přenosu dat pak následuje samotný výpočet přenosové rychlosti. Součástí výsledků pak bude rychlost stahování, odesílání dat, doba odezvy a případně ještě další doplňkové parametry, které jsou použitelné při stanovení kvality služby mobilní sítě, a které lze pomocí mobilního telefonu s operačním systémem Android získat.

Veškeré výsledky jsou na konci měřicího cyklu odeslány ve formátu CSV souboru na vzdálený FTP server, kde jsou využívány k další analýze a vytváření statistik a podobně.

V závěrečné kapitole budou zmíněny některé výsledky a statistiky, které se mi podařily pomocí testovací sondy získat.

1 Mobilní sítě

Mobilní sítě jsou telekomunikační sítě, které jsou charakteristické tím, že uživatel (klient) takové sítě může využívat služeb této sítě pomocí zařízení k tomu určených na různých místech. Tato místa však musí být pokryta odpovídajícím signálem. Na rozdíl od pevných telefonních sítí, jejichž služeb mohou využívat klienti (zákazníci) pomocí zařízení, které jsou k těmto účelům zkonstruovány, pouze ale na určitých konkrétních místech. Tato místa se vyznačují tím, že je k nim přivedeno účastnické vedení typicky ve formě kabelu. K pevným sítím se tedy připojíme pouze tam, kde je přivedeno účastnické vedení, například v určité místnosti v rámci budovy. V mobilních sítích můžeme využívat jejich služeb v místě pokrytí signálem, které je realizováno pomocí základnových stanic, které mohou pokrýt území až několika kilometrů čtverečních.

1.1 Rozdělení mobilních sítí dle způsobu šíření signálu

Prakticky se dají rozdělit mobilní telefonní sítě do tří kategorií dle toho, jakým způsobem je šířen jejich signál a na jakém principu fungují.

1.1.1 Celulární (buňkové) rádiové sítě

Celulární, nebo také buňková rádiová síť je taková telekomunikační síť, u které je signál šířen pomocí určitého množství základnových stanic. Díky dosahu signálů, které šíří jednotlivé základnové stanice, vytvářejí soustavu navzájem se překrývajících buněk. Odtud pochází i pojmenování celulární - z latinského slova "celula", které v překladu znamená buňka. Umístění, typ a další vlastnosti každé ze základnových stanic se řídí určitými pravidly a zvyklostmi. S určitou mírou pravděpodobnosti můžeme předpokládat, že na území, které je méně geograficky členité, má menší množství potencionálních zákazníků k obslužení a nevyskytují se na něm žádné větší vnější vlivy bránící prostupu signálu, jako jsou různé kopce, lesy, výškové budovy a podobně, bude dostatečné pokrytí pomocí menšího množství základnových stanic, které navíc budou moci být od sebe ve větší vzdálenosti. Zatímco v městských podmínkách, kde je velká hustota obyvatelstva a různých vysokých budov bude zapotřebí k dostatečnému pokrytí signálem většího množství základnových stanic.

Provozní frekvence celulárních systémů jsou od 300 MHz do 3 GHz. Spadají tak do části kmitočtového spektra ultra krátkých vln, tzv. UHF (Ultra High Frequency). Tato část kmitočtového pásma se využívá také pro šíření televizního vysílání, pro provoz družicových, leteckých a mobilních rádiových sítí 1G až 4G.

GSM, UMTS i LTE, které jsou předmětem této diplomové práce, patří z hlediska pokrytí mezi bezdrátové rozsáhlé sítě WWAN (Wireless Wide Area Network). Umožňují plně duplexní přenos.

1.1.2 Sítě satelitních telefonů

Jedná se o telekomunikační síť využívající telefony a jiná zařízení, která však nejsou připojena k síti pomocí pozemních základnových stanic, nýbrž pomocí orbitálních telekomunikačních satelitů. Výhodou satelitních telefonů oproti klasickým mobilním telefonům je fakt, že se s nimi můžeme dovolat téměř odkudkoliv z celého světa. Nejsme vázáni na určitou oblast, která je pokryta signálem z jedné ze základnových stanic operátora. Používání takové komunikace však přináší jisté nevýhody a omezení oproti komunikaci skrz buňkové rádiové sítě. Především je tato komunikace dražší. Jednak pořízení satelitního telefonu je nákladnější, tak také jeho následný provoz. Co se týká satelitních telefonů, ty jsou dnes již velikostně i vzhledově srovnatelné s klasickými mobilními telefony, často však vyžadují připojení ještě externí antény. Nevýhodou je, že telefon, respektive externí anténa musí mít přímý výhled na satelit. Komunikace nefunguje například v uzavřených místnostech a budovách. Další omezení tkví v místních zákonech a nařízeních. Především v totalitních zemích je takováto forma komunikace úplně zakázána či velmi omezena. Nejstarší geostacionární síť je Inmarsat, kterou musí ze zákona využívat veškeré lodě plující přes oceán. Další známou sítí je Thuraya. Iridium - síť tvořená nízko letícími satelity - pokrývá i polární oblasti, které nejsou dvěma předešlými sítěmi pokryty.

1.1.3 Sítě bezšňůrových telefonních systémů

Bezšňůrové telefonní systémy naleznou uplatnění především jako součást osobní rádiové komunikace. Zpravidla se tato forma komunikace skládá ze dvou částí. Pevnou část tvoří základna, která se stará o zajištění komunikace mezi pohyblivou (koncovou klientskou) částí a jedním z mnoha druhů dalších sítí. Základna také bývá zároveň nabíjecím zařízením pro pohyblivou část. Ta je často tvořena přenosným telefonem, který je podobný běžně používaným mobilním telefonům nebo je zastoupena miniaturním sluchátkem a podobně. Pokrytí signálem je ve volném prostoru zhruba do 300 metrů, v rámci budov je toto pokrytí v jednotkách až desítkách metrů dle druhu budovy.

V současnosti nejpoužívanějším standardem v této oblasti je DECT (Digital Enhanced Cordless Telecommunications). Standard DECT byl vyvinut a standardizován organizací ETSI (European Telecommunications Standards Institute). Jeho počátky jsou datovány do let 1988 až 1992. Tento systém využívá frekvenční pásmo 1880 MHz až 1900 MHz. V USA je používáno pásmo 1920 MHz až 1930 MHz.

1.2 Rozdělení mobilních sítí dle generace vývoje

Mobilní sítě se dají rozdělit celkem do pěti generací. Zachycují historický vývoj v této oblasti od nejstarší nulté generace až po v současné době nejdokonalejší síť čtvrté generace. V další kapitole se zmíním především o sítích GSM, UMTS a LTE.

1.3 Sítě GSM, UMTS, LTE

Sítě typu GSM, UMTS a LTE jsou součástí řešení této diplomové práce. Budeme se jimi zabývat z hlediska kvality služeb těchto sítí. Především nás bude zajímat měření aktuální přenosové rychlosti a dosažené hodnoty zpoždění v mobilních sítích těchto typů.

1.3.1 GSM

GSM je celulární (buňková) síť pro telekomunikaci. Původně sloužila pouze k přenosu hlasu, později se přidaly další služby založené na přenosu dat. Prvopočátky se datují do roku 1982, kdy byl poprvé vznesen požadavek na standardizovaný celoevropský telekomunikační standard. Technické rysy takové služby byly zveřejněny v roce 1987. Na svědomí je měla skupina CEPT (European Conference of Postal and Telecommunications Administrations). Roku 1990 byla na světě první specifikace systému vytvořená ETSI. V minulosti umožňoval standard GSM datové přenosy CSD (Circuit Switched Data) účtované jako běžný hovor, roku 1997 bylo k dispozici také GPRS připojení účtované dle množství přenesených dat a roku 2003 také "vylepšené" GPRS v podobě EDGE.

Vyhrazené kmitočty jednotlivým typům systémů GSM jsou vypsány v tabulce 1.1. Downlink je směr komunikace od základnové stanice k mobilní stanici, naopak uplink je směr od mobilní stanice k základnové.

Tabulka 1.1: Vyhrazené kmitočty pro systém GSM

Typ GSM	Downlink [MHz]	Uplink [MHz]
PGSM	935 - 960	890 - 915
EGSM	925 - 960	880 - 915
GSM 1800	1805 - 1880	1710 - 1785
GSM 1900	1930 - 1990	1850 - 1910

V systému GSM jsou dva typy kanálů pro komunikaci v systému: logické a signalizační. Logické umožňují přenos uživatelských a řídicích informací před započítím, během a po ukončení hovoru. Signalizační kanály slouží k optimálnímu nastavení mobilního zařízení vůči základnovým stanicím, k handoveru, registraci mobilního zařízení do sítě, signalizaci hovoru a dalším. Na rádiovém rozhraní je použita modulace GMSK (Gaussian Minimum Shift Keying). Více informací o architektuře a fungování sítí GSM je k dispozici v seznamu literatury [1].

1.3.2 UMTS

UMTS sítě jsou evolucí GSM sítí. Vznikly na základě dohody o spolupráci 3GPP vytvořené na konci roku 1998. První verze UMTS byly dostupné roku 2000. V Evropě je vyhrazené frekvenční pásmo 2110 - 2170 MHz pro downlink a 1920 - 1980 pro uplink.

Oproti GSM, kdy bylo použito GMSK modulace, je nyní použito vyspělejší QPSK (Quadrature Phase Shift Keying) modulování. Šířka kanálu je 5 MHz. Bližší informace o sítích UMTS se můžete dočíst v knize uvedené v seznamu literatury [2].

1.3.3 LTE/EPC

Projekt LTE započal v roce 2004 pod vedením 3GPP. Jedná se o evoluci dřívějších UMTS technologií. Technologie LTE se zaměřuje především na vysokorychlostní připojení k Internetu. Podporuje služby využívající přepojování paketů (PS) včetně VoIP. Dosažitelné rychlosti LTE jsou až 300 Mb/s pro stahování a 75 Mb/s pro odesílání. LTE využívá jak časově (TDD), tak i frekvenčně (FDD) dělený duplex.

Na fyzické vrstvě je nově použito OFDM (Orthogonal Frequency Division Multiplexing) multiplexování pro downlink a SC-FDMA (Single Carrier - Frequency Division Multiple Access) pro uplink. LTE standard umožňuje měnit šířku frekvenčního pásma v rozmezí 1,25 MHz až 20 MHz a také využívat technologii MIMO (Multiple Input Multiple Output), která dovolí použití více antén pro vysílání a příjem. Díky tomu se zvětší přenosová kapacita, zvýší se přenosové rychlosti. Podle IMT-2000 jsou k dispozici technologii LTE/EPC tyto frekvenční rozsahy: 450-470 MHz, 698-960 MHz, 1710-2025 MHz, 2110-2200 MHz, 2300-2400 MHz, 2500-2690 MHz a 3400-3600 MHz. Další informace o sítích LTE jsou k dispozici v seznamu literatury [3]. Mapu pokrytí území České republiky jednotlivými operátory a jim přidělenými frekvenčními pásmy lze pak dohledat na webu ČTÚ [4].

2 Kvalita služby pro mobilní technologie

Pod pojmem kvalitní si můžeme představit stav, kdy výrobek, nebo v našem případě služba, má kladné vlastnosti. Chová se a funguje tak, jak by měla. Je tak v souladu se specifikací, kterou prodejce avizuje a zároveň kterou zákazník požaduje a očekává. Dle Akademického slovníku cizích slov je kvalita "souhrn užitečných vlastností výrobku nebo služby, souhrn typických, zpravidla kladných vlastností."

S kvalitou služby úzce souvisí pojem QoE (Quality of Experience). Jedná se obvykle o procentuální vyjádření množství zákazníků, kteří jsou s určitou poskytovanou službou spokojeni. Zahrnuje jak subjektivní, tak objektivní hodnocení služby ze strany zákazníků a uživatelů. Aby bylo toto hodnocení pokud možno co nejvíce vypovídající a nej přesnější, je nutno dotázat se na zkušenost a spokojenost s danou službou co největšího množství uživatelů. Každý uživatel totiž může vnímat a hodnotit stejnou situaci trochu jinak, proto je nutné tato hodnocení nějak zprůměrovat a zpřesnit. Z hlediska hodnocení služby poskytnuté konkrétním uživatelem je rozhodující především cena služby, dostupnost, spolehlivost, bezpečnost, soukromí a srozumitelnost. Pro každou společnost je důležité, aby měla hodnocení svých produktů a služeb pokud možno co nejlepší. Nejen z marketingového hlediska je žádoucí mít co nejvíce spokojených zákazníků, kteří budou využívat služeb určité firmy a tím ji dále propagovat a dělat jí reklamu.

V oblasti mobilních technologií je kvalita služby určena především dostupností, rychlostí, zpožděním a v neposlední řadě také cenou. Dostupnost služby je dána mírou pokrytí území signálem mobilního operátora. V současnosti je míra pokrytí v naší republice vysoká. Signálem je pokryto téměř celé území všemi operátory. Menší výjimky jsou spíše ve velmi řídké obydlených či neobydlených oblastech, jako jsou hluboké a rozsáhlé lesy, na horách a podobně. Dokonce jsou signálem pokrývány i dříve nepokryté oblasti, jako jsou hypermarkety, vlaky, metro, tunely. Nejmodernějšími technologiemi, jako je LTE, je území republiky teprve pokrýváno. V současné době, kdy se využívání služeb mobilních technologií stále více zaměřuje na datové přenosy, je pořadí důležitější rychlost stahování a odesílání dat skrze mobilní síť. Každý by chtěl mít pokud možno co nejrychlejší datové přenosy ve svém zařízení a mít tak co nejkomfortnější stahování či odesílání dat. Dalším důležitým parametrem k určení kvality služby mobilní technologie je doba zpoždění signálu. Zpoždění signálu je v mobilních sítích způsobeno nedostatečně rychlým zpracováním přenášených dat nebo fyzikálním omezením rychlosti šíření daného signálu. V pevných IP sítích je doba zpoždění typicky v desítkách milisekund, v mobilních sítích je to pak až v řádech stovek milisekund. Zpoždění je v mobilních sítích způsobeno především díky zpracování signálu klientskou mobilní stanicí (díky procesu kódování a komprese) a také v rámci procedury přenosu rádiového signálu. Dále může být způsobeno nedostatečným technickým zázemím mobilního operátora či technickými problémy na trase přenosu komunikace. Neméně závažný je tzv. jitter. Ten zachycuje rozptyl, rozkolísání zpoždění. Především u tzv. realtime služeb, kdy dochází ke zpracování v reálném čase (například videohovor, online stream videa nebo zvuku), má tento parametr velký dopad a vliv. Tyto služby totiž očekávají periodický příchod paketů ve správném pořadí. U přenosu hlasu je

téměř jedno, zdali nějaké pakety dorazily s různou mírou zpoždění nebo nedorazily vůbec. Obojí má za následek stejný efekt, a to zhoršení kvality hovoru. Ke korekci těchto jevů je využíváno různých bufferů, které mají za úkol minimalizovat tyto vlivy. Jak již bylo zmíněno, je důležitá také cena služby. Koncový zákazník se musí dobře seznámit s podmínkami služby, kterou si objednává. Většinou totiž jako v ostatních odvětvích platí přímá úměrnost. Čím více si za službu zaplatíme, tím také můžeme očekávat větší kvalitu dané služby a naopak. U mobilních technologií toto platí obvykle také.

Měření kvality služby mobilních technologií provádějí všichni mobilní operátoři. Je to totiž pro ně velmi důležitý ukazatel toho, jak v daném místě fungují jimi poskytované služby, umožňuje jim to reagovat na skutečný stav sítě a potřeby pro další vývoj a údržbu sítě. Takové měření kvality je procesem dlouhodobým a kontinuálním. Pravidelně se provádí v rámci míst, kde jsou poskytovány služby mobilních operátorů. Výsledky měření slouží pro operátory jako podklad k tomu, aby mohli správně nastavit vysílací výkony příslušných základnových stanic a tím upravit pokrytí signálem v daných oblastech, dále při rozhodování u procesu výběru nejvhodnější základnové stanice k obsluze koncové mobilní stanice a tím umožnění tzv. handoveru (proces, při kterém je pro komunikaci mezi koncovou mobilní stanicí a mobilní sítí vybrána základnová stanice případně vybrán kanál s nejkvalitnější možností spojení), k optimalizaci sítě a v neposlední řadě takto mohou být odhaleny různé další problémy, které nejsou jiným automatizovaným způsobem zjištěitelné.

2.1.1 Nástroje k měření přenosových parametrů

K měření přenosových parametrů mobilních sítí se používají profesionální měřicí přístroje výrobců, jako jsou Ascom, Rohde & Schwarz, Agilent. Tyto přístroje jsou profesionální, nabízí mnoho možností a parametrů k měření.

Ascom Holding AG - telekomunikační společnost založená ve Švýcarsku v roce 1987. Soustřeďuje se na bezdrátová řešení, testování a optimalizaci bezdrátových mobilních sítí. Nyní má asi 1600 zaměstnanců. Svá řešení dodává především institucím, jako jsou nemocnice, továrny, vězení, hotely apod. Část firmy, která se zabývá měřením, analýzou a optimalizací mobilních sítí, se nazývá Ascom Network Testing. Do jejich portfolia spadá značka TEMS. Ta zajišťuje přístroje a programy pro telefony, počítače a další zařízení sloužící k testování, monitoringu a analýze mobilních sítí. Základem systému TEMS je zařízení, kterým se měří kvalita mobilní sítě pomocí navazování hovorů, odesílání SMS či datové komunikace. Díky těmto procesům pak může vyhodnotit kvalitu například na základě množství neuskutečněných hovorů, množství náhle přerušovaných hovorů, měření kvality hlasu či díky měření rychlosti přenosu dat a zpoždění. Jedním z nabízených produktů je TEMS Pocket (zobrazen na obrázku 2.1). Jedná se o software, který po instalaci do kompatibilního mobilního telefonu z něj udělá měřicí stanici. Cena nástroje TEMS Pocket včetně mobilního telefonu se pohybuje v řádech desítek tisíc korun.



Obrázek 2.1: Tems Pocket (zdroj: <http://www.dhgate.com/>)

Rohde & Schwarz - telekomunikační společnost založená v Německu v roce 1933. V současné době má zhruba 9300 zaměstnanců po celém světě a zabývá se mnoha oblastmi, týkajícími se hlavně bezdrátových sítí a komunikace. Nabízí řešení týkající se správy, údržby a monitoringu bezdrátových sítí, jejich zabezpečení, podporu, servis a systémovou integraci svých řešení nejen pro mobilní operátory. Společnost nabízí mnoho různých přístrojů určených pro testování nejen mobilních bezdrátových sítí. Přístroj Rohde & Schwarz CMW500 (vyobrazen na obrázku 2.2) je komplexním nástrojem pro analýzu sítí typu GPS, DVB-T, WiFi, mobilní sítě 2G, 3G a další. Jeho orientační cena se blíží milionu korun českých.



Obrázek 2.2: Rohde & Schwarz CMW500 (zdroj: <http://www.rohde-schwarz.cz/>)

Agilent - firma zabývající se nejen telekomunikacemi založená v USA společností Hewlett-Packard v roce 1999. V současnosti má kolem 20 500 zaměstnanců. Působení společnosti spadá do oblastí biotechnologie, elektroniky a chemického průmyslu. Pro účely testování bezdrátových mobilních sítí zmíním například přístroj Agilent E6607B EXT Wireless Communications Test Set (zobrazen na obrázku 2.3). Tento přístroj zvládá testování a analýzu 2G, 3G, LTE, Bluetooth, GPS a dalších druhů bezdrátových sítí. Cena tohoto přístroje se pohybuje kolem 700 000,- Kč.



Obrázek 2.3: *Agilent E6607B EXT (zdroj: <http://www.keysight.com/>)*

Ze zde zmíněných a vyobrazených příkladů komplexních profesionálních nástrojů k analýze a testování mobilních sítí vyplývá, že se opravdu jedná o často neprakticky velké přístroje, které jsou navíc velmi drahé a poskytují velké množství funkcí, které ke splnění našeho úkolu nepotřebujeme.

2.1.2 Parametry určující kvalitu služby

Nás budou zajímat hlavně dva parametry týkající se kvality služby mobilní technologie GSM, UMTS a LTE. První z parametrů je přenosová rychlost. Ta udává, jaké množství dat je přeneseno za jednotku času. Základní jednotkou přenosové rychlosti je bit/s (bit za sekundu). Dále se používají násobky - Kbit/s (kilobit za sekundu), Mbit/s (megabit za sekundu). Někdy se také používá jednotka B/s (bajt za sekundu) a její násobky. Platí vztah, kdy $1 \text{ B/s} = 8 \text{ b/s}$. Rychlost připojení k Internetu se uvádí jako garantovaná, nebo agregovaná. U garantované rychlosti připojení máme jistou minimální hodnotu této rychlosti, pod níž se nikdy nesmí dostat. Připojení k Internetu s garantovanou rychlostí je určeno zejména pro ISP, firmy, servery a podobně a je většinou mnohem dražší oproti agregovanému připojení. To je zpravidla levnější, ale znamená to, že o rychlost se dělí větší množství klientů. Rychlost připojení se dynamicky mění podle aktuálního vytížení sítě. V mobilních sítích se vyskytuje především ono připojení agregované s proměnlivou negarantovanou rychlostí. Měření aktuální přenosové rychlosti se děje za pomoci speciálních měřicích serverů, které jsou k tomuto účelu zkonstruované. V našem případě si takové měření rychlosti naprogramujeme sami. Bude založeno na principu, kdy

nejdříve bude odesíláno po určitou dobu množství dat z mobilního telefonu na vzdálený server. Podle množství dat, které se za předem určenou dobu odešle, vypočteme aktuální rychlost odesílání. Při stahování dat ze vzdáleného serveru do mobilního telefonu opět stejným způsobem určíme aktuální rychlost stahování dat. Na takovém způsobu fungování jsou založeny i již zmiňované servery specializující se na měření rychlostí připojení. Průměrná rychlost stahování dat v českých mobilních sítích je v listopadu 2014 dle výsledků serveru www.dsl.cz následující: GPRS 41 kbit/s, EDGE 112 kbit/s, 3G 6 945 kbit/s, LTE 22 594 kbit/s. Druhým z pro nás nejzásadnějších parametrů určujícím kvalitu služby mobilní technologie je zpoždění (jiným slovem latence). To je dáno fyzikálním omezením rychlosti šíření signálu a zpracováním při procesu přenášení dat na přenosové trase mezi účastníky komunikace. V našem případě budeme k určení aktuální hodnoty zpoždění využívat program ping. Ten využívá protokol ICMP. Umožňuje zjistit délku zpoždění tím, že v pravidelných intervalech odesílá na předem definovanou IP adresu nebo server IP datagramy a od vzdálené strany čeká na odezvu. Podle časového intervalu od odeslání IP datagramu do přijetí odpovědi určí dobu zpoždění. Délka latence do jisté míry určuje kvalitu služby tím, že čím je vyšší latence, tím je uživatelův požitek z dané služby nižší. Vysoká hodnota latence omezuje možnosti využívání některých služeb nebo tyto možnosti vylučuje úplně. Týká se to především služeb náchylných na délku latence, jako jsou například online hry a VoIP telefonie. Obvyklé hodnoty zpoždění jsou u 2G větší než 200ms, u 3G se nachází v rozmezí 40ms - 200ms a u LTE jsou menší než 40ms.

Další možností stanovení kvality služby mobilních sítí jsou MOS (Mean Opinion Score) a BER (Bit Error Rate). Parametr MOS je popsán v doporučení P.800 vydaném ITU-T v srpnu roku 1996. Jedná se o subjektivní hodnocení kvality hovoru, které provádí jeden nebo více lidí v laboratorním prostředí a kdy je srovnávána kvalita probíhajícího hovoru s kvalitou analogového spojení. Toto hodnocení tedy není objektivní, je nákladné a nemá příliš vypovídající hodnotu. Rozsah hodnocení je na stupnici 1 (nejhorší kvalita hovoru) až 5 (nejlepší kvalita hovoru), jak lze vyčíst z tabulky 2.1:

Tabulka 2.1: *Stupnice hodnot parametru MOS, tzv. stupnice poslechové kvality*

MOS	Kvalita hovoru	Slovní popis
5	Výborná	Bez rušení, vynikající komunikace
4	Dobrá	Zanedbatelné rušení, přirozená komunikace
3	Střední	Znatelnější rušení, vyžaduje větší úsilí při komunikaci
2	Špatná	Značné rušení, komunikace je velmi obtížná
1	Nedostatečná	Řeči nelze rozumět, nevyhovující komunikace

Parametr BER je již více objektivním ukazatelem. Jedná se o bitovou chybovost, která je vyjádřena poměrem chybně přijatých bitů k celkovému množství přijatých bitů za určité časové období. Obvykle se udává v procentech a značí, jaké množství bitů bude muset být kvůli chybě přeneseno znovu.

$$BER = \frac{b_E}{v \times t} \quad \left[-; bit; \frac{bit}{s}; s \right]$$

b_E ... přijaté bity; v ... přenosová rychlost; t ... doba měření

Rovnice (2.1)

3 Testování a optimalizace mobilních sítí

Testování mobilních sítí, především měření mobilních datových sítí, lze rozdělit do několika hlavních částí:

- **Drive testování**

Drive testování probíhá v rámci sítí, které jsou už v ostrém provozu. Slouží především k jejich optimalizaci. Testování probíhá průběžně během dne a dle analýzy naměřených výsledků se realizuje optimalizace. Tento způsob tedy slouží ke zkvalitnění poskytovaných služeb a k porovnání poskytovaných služeb s konkurencí.

Operátor T-Mobile v České republice používá systémy TEMS Symphony od firmy Ascom. Dva systémy jsou umístěny v Praze a jeden v Brně. Dále jsou využívány sondy TEMS Automatic v počtu asi 10 kusů pro celou republiku.

- **Optimalizace sítě, řešení zákaznických zkušeností**

Pro řešení zákaznických zkušeností, optimalizaci sítě a testování mobilní sítě na konkrétních místech jsou využívány měřicí systémy Romes od společnosti Rohde & Schwarz a TEMS Investigation od firmy Ascom.

- **Monitorování síťových statistik**

K monitorování síťových statistik a získání tak lepších informací o fungování a využití sítě je využíván systém NIMS-PrOptima od firmy Mycom.

Celkem se touto problematikou týkající se testování, monitorování a optimalizací mobilní sítě zabývá v rámci mobilní sítě T-Mobile Czech Republic a.s. asi 20 lidí, kteří jsou rozdělení do několika menších týmů po celé České republice. Veškeré informace získané pomocí testování a monitorování mobilních sítí jsou pak využity k optimalizaci mobilní sítě, snížení nákladů a zkvalitnění poskytovaných služeb zákazníkům.

4 Použité technologie

Při řešení diplomové práce bylo využito mobilní zařízení s operačním systémem Android, které provádí veškerá měření a získané výsledky odesílá na vzdálený server. Nedílnou součástí mého řešení je také zařízení zkonstruované pomocí platformy Arduino, které mobilní telefon hlídá a udržuje jej v optimálním provozuschopném stavu.

4.1 Android

Android je operační systém založený na Linuxovém jádře. Vyvíjen je pod vedením společnosti Google, potažmo uskupením firem nazvaným Open Handset Alliance, které v současnosti sdružuje 84 firem z celého světa. Android je primárně určen jako operační systém pro dotykové mobily a tablety, postupně se však stává součástí chytrých televizí, hodinek, automobilů, digitálních fotoaparátů, herních konzolí a počítačů. Android je nejrozšířenější mobilní operační systém. Ve druhém čtvrtletí roku 2014 tvořilo celosvětově 84,7% všech prodaných chytrých telefonů právě zařízení s OS Android. V říjnu 2008 se začíná prodávat první komerční telefon s Androidem - HTC Dream. Nejnovější verze v době psaní této práce je Android verze 5.0.1 Lollipop, který byla vydán 2. prosince 2014.

Architektura Androidu sestává z pěti vrstev, které se navzájem doplňují. Jedná se o nejnižší vrstvu Linux Kernel. Jak už název napovídá, jde o jádro operačního systému postavené na Linuxu. Druhou vrstvou jsou knihovny napsané v programovacích jazycích C/C++, které jsou využívány různými součástmi systému. Třetí vrstva je Android Runtime. Ten je tvořen DVM (Dalvik Virtual Machine). Virtuální stroj DVM je použit ke spouštění jednotlivých aplikací, kdy překlad každé aplikace probíhá následovně: Nejdříve dojde ke zkompileování Java kódu do Java byte kódu, který je následně pomocí Dalvik kompilátoru překompileován do výsledného Dalvik byte kódu. Čtvrtou vrstvou je Application framework, který je z hlediska vývojářů nejdůležitější. Stará se totiž o přístup k jednotlivým prvkům uživatelského rozhraní, k uloženým datům, k hardwaru a dalším službám a funkcím systému. Nejvyšší vrstva je aplikační. Ta sestává z předinstalovaných nebo dodatečně doinstalovaných aplikací. Například může jít o organizační aplikace, kontakty, SMS, emailový klient, přehrávače, hry a další aplikace.

Vývoj aplikací je oficiálně skrze vývojové prostředí Eclipse. K ulehčení programování pro Android je k dispozici SDK (Android Software Development Kit). Ten obsahuje všechny nezbytné nástroje k optimálnímu návrhu, realizaci a testování aplikací. To může probíhat pomocí emulátoru, který je součástí SDK.

Oficiálně jsou volně dostupné i placené verze programů k dispozici přes tzv. Google Play. Jedná se o službu provozovanou společností Google, která nabízí prodej a šíření aplikací registrovaným uživatelům. Obdoba internetového obchodu.

Výhodou operačního systému Android je jeho otevřenost. V případě získání administrátorského (tzv. root) oprávnění může být systém upraven a přizpůsoben k obrazu

svému. Velké množství lidí vlastníci zařízení s tímto systémem vede ke spoustě dostupných návodů, postupů, rad a také aplikací a služeb. Z velkého rozšíření a oblíbenosti pak bohužel také plynou jistá bezpečnostní rizika, kdy se hackeři a kybernetičtí zločinci zaměřují na tuto platformu. Velmi zajímavá kniha o úpravách systému Android, o získávání administrátorského přístupu, vytváření vlastních skriptů a podobně se nazývá *Android Hacker's Toolkit* [5].

4.2 Arduino

Arduino je otevřená platforma založená na mikrokontrolerech ATmega od firmy Atmel. Arduino se používá k vytváření vlastních samostatných zapojení či k připojení k počítači, kdy jeho služeb může využívat externí software běžící na počítači. Arduino vzniklo v roce 2005 v Itálii. Zpočátku byl projekt plánován jako edukační pomůcka k vytváření prototypů pro studenty, vysoké školy a další instituce. Později se však rozšířil masově i mezi technické nadšence. Pro své univerzální využití je oblíben také mezi běžnými domácími kutily.

Arduino hardwarová část je založená na ATmega mikrokontrolerech. Možná jsou rozšíření o velké množství dalších komponent rozšiřujících funkčnost, jako jsou LED diody, displeje, senzory, tlačítka, moduly pro bezdrátovou i drátovou komunikaci a jiné. Tím se nabízí nepřeberné množství využití. V současnosti nejoblíbenější deska Arduino Uno R3 obsahuje ATmega328 mikrokontroler. K dispozici je 14 digitálních vstupně/výstupních pinů, 6 analogových vstupů, USB připojení, resetovací tlačítko, 32 KB paměti flash, 2 KB SRAM, 1 KB EEPROM a operuje na frekvenci 16 MHz. Napájen může být skrz USB připojení nebo pomocí externího zdroje o napětí 5V.

Vlastní aplikace jsou psány pomocí jazyka podobného C/C++, vychází z jazyka Wiring. Díky faktu, že hlavní mikrokontroler již obsahuje bootloader starající se o základní nastavení mikrokontroleru, odpadá programátorovi tato povinnost. Vývoj aplikací probíhá pomocí Arduino IDE programovacího prostředí.

Hlavní výhody tkví v jednoduchosti, platformní nezávislosti a množství dostupných návodů, které jsou k dispozici například na oficiálních stránkách [6].

4.3 Mobilní telefon Sony Xperia V

Jako mobilní zařízení, které bylo zvoleno k vývoji měřicí aplikace, byl zvolen mobilní telefon Sony Xperia V, někdy také označován Sony Xperia LT25i. Sony Xperia V je pro účely vývoje testovací sondy více než dostačující. Telefon disponuje těmito parametry:

Tabulka 4.1: *Parametry a údaje o přístroji*

Vlastnost	Hodnota
Rozměry	129 x 65 x 10,7 mm
Hmotnost	120 g
Pásmo GSM	850, 900, 1 800, 1 900 MHz
Pásmo 3G	850, 900, 2 100 MHz
Pásmo 4G	800, 850, 1 800, 2 100, 2 600 MHz
GPRS, EDGE	ano
UMTS/HSPA	42,2 / 5,76 Mb/s
LTE	100 / 50 Mb/s
Paměť RAM	1 024 MB
Uživatelská paměť	8 000 MB
Procesor	Qualcomm Snapdragon S4 Plus (1 500 MHz)
Architektura procesoru	ARM v7-A
Operační systém	Android verze 4.3



Obrázek 4.1: *Mobilní telefon Sony Xperia V*

5 Vlastní řešení

Tato kapitola se zabývá samotným řešením zadání diplomové práce. Obsahuje kompletní informace o tom, jak jsem řešil veškeré úkoly nutné ke splnění zadání a proč bylo zvoleno zrovna takové řešení.

Veškeré měření, zjišťování doplňkových údajů týkajících se přenosových parametrů v mobilní síti a další funkcionalita je řešena pomocí skriptů. Tyto úkony jsou prováděny v rámci mobilního telefonu s operačním systémem Android, který vychází z jádra Linuxu. Proto je umožněno na mobilním telefonu spouštět skripty podobné těm, které jsou běžně používané například na desktopových či serverových strojích s operačními systémy taktéž vycházejícími z jádra Linuxu. Tento způsob řešení se ukázal být tím neoptimálnějším a nejpoužitelnějším v praxi. Můžeme totiž přistupovat k proměnným a dalším zdrojům přímo v rámci operačního systému, můžeme náš skript automaticky spouštět v určený čas nebo po určité události, například jako v našem případě po restartování telefonu. O tom, jak probíhá samotné měření, se budu detailněji zmiňovat v dalších částech diplomové práce.

Zpočátku byl vývoj měřicího skriptu prováděn na desktopovém počítači v prostředí operačního systému Ubuntu. To však byly pouze prvopočátky vývoje aplikace. Už v raných fázích vývoje se totiž ukázalo, že sice Android vychází z Linuxového jádra a umožňuje podobný chod a chování systému, nicméně téměř ve všech oblastech s různými odlišnostmi. Na tyto odlišnosti v použití a syntaxi i základních příkazů jsem přišel hned ve chvíli, kdy jsem první náznaky skriptu vyvinutého na desktopu zkusil spustit na mobilním telefonu. V lepším případě některé příkazy měly jen jinou syntaxi nebo parametry, v tom horším případě některé příkazy nebyly na Androidu dostupné vůbec a muselo se hledat jiné řešení. Zpočátku také veškerá měření probíhala v rámci přenosu přes síť WiFi. Neměl jsem k dispozici testovací SIM kartu s neomezenými datovými přenosy. Pro následné testování datových přenosů v mobilních sítích jsem poté musel skript značně upravovat, nicméně jsem měl již předpřipravenou kostru a znal způsob, jakým vše zrealizovat.

Stěžejní částí diplomové práce bylo zrealizovat monitorovací systém přenosových parametrů v mobilních sítích. Aby tento systém mohl pracovat autonomně, samočinně a spolehlivě, bylo nutné vymyslet funkci "watchdog" pro tuto měřicí sondu. Mobilní telefon, operační systém nebo některá z jeho částí, případně samotná měřicí aplikace by se mohla časem začít chovat neočekávaně, například se zaseknout. Proto byla zrealizována funkce "watchdog" měřicí sondy pomocí otevřené elektronické platformy Arduino. Platforma Arduino byla vybrána pro svou rozšířenost, spolehlivost a další užité vlastnosti. Díky ní můžeme v pravidelných časových intervalech preventivně restartovat telefon a předcházet tak dlouhodobějším výpadkům ve fungování sondy, které by mohly být způsobené například chybou operačního systému telefonu či některé z jeho součástí. O konkrétní implementaci Arduina včetně popisu fungování všech jeho použitých částí se budu rozepisovat v některé z následujících částí diplomové práce.

5.1 Měřicí aplikace – úvod, skript *init*

Měřicí aplikace sestává z několika skriptů. Tyto skripty díky vhodnému nakonfigurování pracují plně samostatně. První ze skriptů - *init* - se nachází v telefonu ve složce */etc/init.d/*. Tento skript má na starost především automatické spuštění měřících procesů po nastartování operačního systému. Skript *init* volá po nastartování systému další ze skriptů - *test.sh*. Ten se může nacházet ve vnitřním úložišti mobilního telefonu, případně na externím úložišti v podobě paměťové karty. V praxi se ukázalo použití externího úložiště jako vhodnější. Skript *test.sh*, který je v ideálním případě uložen ve složce */sdcard/*, provádí veškeré měření a zjišťování všech potřebných parametrů. Další doplňkové informace jsou pak získány pomocí skriptu *laccellid.sh*, který se opět v ideálním případě nachází na externí paměťové kartě, čili ve složce */sdcard/*. Jeho úkolem je získání aktuálních hodnot LAC (Local Area Code) a Cell ID. Tyto údaje slouží k bližšímu určení toho, kde bylo měření provedeno a oproti jaké základnové stanici.

Inicializační skript *init* se nachází ve složce */etc/init.d/*. Tato složka je význačná tím, že skripty v ní umístěné jsou automaticky spouštěny po nabootování operačního systému Android. Není to však pravidlem pro veškeré verze operačních systémů Android. Tento způsob automatického spouštění skriptů umístěných ve složce */etc/init.d/* je defaultně nastaven především pro operační systémy, které jsou upraveny uživatelskou komunitou na Internetu (např. populární CyanogenMod), případně pro operační systémy upravené přímo výrobcí mobilních telefonů (především výrobci z Číny, jako je například Xiaomi).

V našem případě, kdy probíhal vývoj měřicí aplikace na mobilním telefonu Sony Xperia V s operačním systémem Android ve verzi 4.3, jsem musel automatické spuštění skriptů po startu mobilního telefonu umístěných ve složce */etc/init.d/* nejdříve zprovoznit. V defaultním stavu a nastavení systému toto nefungovalo automaticky. Proces zprovoznění požadovaného chování složky */etc/init.d/* je naštěstí jednorázový a dá se realizovat v podobě spuštění skriptu. Tento skript má přes 200 řádků kódu a musí mít před spuštěním nastavena práva pro spuštění pomocí příkazu *chmod*. Pro úspěšné fungování skriptu jej musíme spustit pod tzv. *root* administrátorským účtem. Pokud vše proběhne dle očekávání, máme po několika málo minutách, které navíc obsahují dvě restartování mobilního zařízení, plně funkční a připravené automatické spuštění skriptů po restartování mobilního telefonu - první předpoklad pro naše automatizované a plně samočinné měření přenosových parametrů mobilních sítí.

Kvůli chování systému Android, kdy dochází při restartování telefonu k úpravám práv souborů a dalším modifikacím v souborovém systému, které mají udržovat operační systém v původním, tj. funkčním stavu, je součástí skriptu obranný mechanismus, který při startování operačního systému nastaví správná přístupová práva pro složku */etc/init.d/* včetně jejího obsahu (všech automaticky spouštěných skriptů).

Dalším problémem byl fakt, že náš měřicí skript měří a monitoruje stav mobilního zařízení po dobu několika minut, než se mu podaří posbírat veškerá požadovaná data a údaje. Nicméně mobilní telefon po nastartování do provozuschopného stavu, tj. po několika desítkách sekund až jednotkách minut podle typu mobilního telefonu a operačního systému, přejde do

fáze úsporného režimu, kdy se zhasne displej a především se přeruší vykonávání všech aplikací a procesů, které nejsou pro základní fungování telefonu nezbytně nutné. Základním fungováním telefonu v této fázi se rozumí mobilní zařízení, které naslouchá přijetí hovoru, krátké textové zprávy SMS a dalším. Udrží se ve stavu pohotovosti. Vykonávání dalších aplikací a služeb, jako jsou hry, WiFi připojení a jiné, které nejsou důležité a mohly by způsobit úbytek energie, je pozastaveno. Tyto aplikace a služby jsou “zmrazeny“. Stejným případem je i vykonávání našeho měření. V lepším případě se alespoň měření započne a přeruší se v době vypnutí displeje a usnutí mobilního zařízení. V tom horším případě se měření ani nezačne. Ani jeden z těchto případů ale není žádoucí. Výsledky měření by mohly být velmi zkreslené nebo by nemusely být žádné. Proto jsem objevil dva způsoby, jak tomuto chování předejít. První způsob úplně zabrání telefonu přejít do úsporného režimu. Využívá fakt, že měřicí sonda je neustále připojená na nabíjecím zařízení. Pomocí volby v nastavení, která je dostupná v *Nastavení -> Možnosti pro vývojáře -> Nevypínat obrazovku*. Pokud tuto položku zaškrtneme, obrazovka se při nabíjení nepřepne do režimu spánku. Po celou dobu fungování bude mít mobilní zařízení zapnutý displej a bude tak v aktivním stavu. Druhou možností, která je elegantnější a navíc šetří energii i displej mobilního telefonu, je použití tzv. *wake locku*. Volně by se tento výraz dal přeložit jako zámek proti usnutí. Jedná se o mechanismus, díky kterému můžeme indikovat potřebu naší aplikace (skriptu) zůstat aktivní. Zamezíme tak přejít do úsporného režimu. Na začátek inicializačního skriptu *init* stačí napsat řádek, který provede inicializaci nového *wake locku*:

```
echo temporary > /sys/power/wake_lock
```

a na konci měření použijeme příkaz pro zrušení námi nastaveného *wake locku* tak, aby se zbytečně nespotřebovala elektrická energie a zdroje systému:

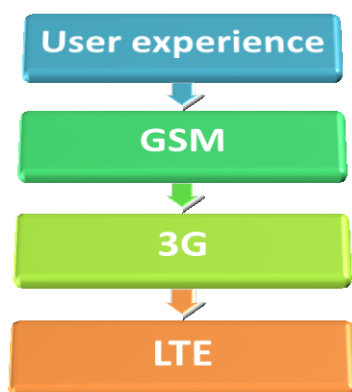
```
echo temporary > /sys/power/wake_unlock
```

Tyto příkazy se postarají o zabránění telefonu v přechodu do úsporného režimu po dobu měření přenosových parametrů, které vykonávají skripty. Více informací o *wake lock* je k dispozici v seznamu použité literatury [7].

Součástí inicializačního skriptu je také ochranný časový interval. Tento časový interval slouží jako ochranná doba, která dá operačnímu systému telefonu dostatečný čas k plnému nastartování, připojení se na mobilní síť a nastartování veškerých služeb. V případě testovacího zařízení Sony Xperia V s operačním systémem Android ve verzi 4.3 se ukázal být jako dostatečný časový interval 90 sekund. Teprve po uplynutí tohoto časového intervalu jsou nejprve nastavena příslušná přístupová práva ke spuštění skriptů a poté také volán samotný skript *test.sh*, který se nachází ve složce */sdcard/*. Tím je mu předáno řízení ve vykonávání měření přenosových parametrů mobilních sítí.

5.2 Měřicí skript test.sh

Nejdůležitější roli v rámci celé měřicí aplikace zastává právě skript *test.sh*. Ten je volán inicializačním skriptem *init* 90 sekund po nastartování operačního systému. Skládá se z příkazů, které jsou nutné pro uskutečnění měření, získání výsledků a jejich následné odeslání na vzdálený FTP server k dalšímu využití. Testovací skript provádí měření a zjišťování dalších parametrů mobilních sítí v pořadí: 1. Defaultní nejvhodnější síť po zapnutí mobilního telefonu (tzv. user experience) -> 2. GSM síť -> 3. 3G síť -> 4. LTE síť, jak lze vidět na obrázku 5.1.



Obrázek 5.1: Sekvence měření jednotlivých typů sítí

5.2.1 Přepínání typů mobilních datových sítí

K přepínání mezi jednotlivými typy mobilních sítí slouží program *sqlite3*. Jedná se o program fungující na bázi příkazové řádky, který slouží k řízení SQLite databázi vytvořených operačním systémem Android. Nástroj *sqlite3* zahrnuje mnoho užitečných příkazů. Například příkaz *.dump* slouží k výpisu obsahu tabulek, příkaz *.schema* slouží k zobrazení struktury existující tabulky. Díky tomuto nástroji můžeme vytvářet nové databáze nebo upravovat databáze stávající. Využívat *sqlite3* můžeme vzdáleně pomocí nástroje *adb* spuštěného na počítači s připojeným mobilním zařízením, nebo lokálně přímo v rámci terminálu spuštěného na mobilním zařízení. Většina mobilních telefonů, se kterými jsem se setkal, měla tento nástroj již defaultně nainstalován. Pokud však chybí, je snadné jej do telefonu doinstalovat. K tomuto účelu slouží například program *SQLite Installer* zdarma dostupný z Google Play. Nebo si můžeme *sqlite3* doinstalovat sami, jednoduše stažením souboru *sqlite3* z Internetu a jeho následným zkopírováním do složky */system/xbin/*. Veškeré SQLite3 databáze jsou uloženy ve složce */data/data/<název_balíčku>/databases*. Nás konkrétně bude zajímat databáze: */data/data/com.android.providers.settings/databases/settings.db*. Zde jsou uložena některá nastavení operačního systému, která Android využívá. Databáze sestává z těchto šesti tabulek:

- android_metadata
- bluetooth_devices
- bookmarks
- global
- secure
- system

Pro nás je nejdůležitější tabulka *global*. Konkrétně její položka *preferred_network_mode*. Tato položka obsahuje informaci, k jakému typu sítě se má mobilní telefon připojovat. Celý příkaz ke změně hodnoty tohoto parametru je následující:

```
sqlite3  
/data/data/com.android.providers.settings/databases/settings.db  
"update global SET value=9 WHERE name='preferred_network_mode'"
```

Parametrem "*value=9*" nastavíme konkrétní preferovaný režim mobilní sítě. Možnosti, které můžeme nastavit pomocí hodnoty *value* parametru *preferred_network_mode* jsou tyto:

Tabulka 5.1: Seznam možných hodnot parametru *preferred_network_mode* [8]

Hodnota value	Význam hodnoty
0	WCDMA preferováno
1	Pouze GSM (pouze 2G sítě)
2	Pouze WCDMA (pouze 3G sítě)
3	GSM automaticky
4	CDMA automaticky
5	Pouze CDMA
6	Pouze EvDo
7	Automaticky GSM/CDMA
8	Automaticky LTE/CDMA
9	Automaticky LTE/GSM
10	Automaticky LTE/GSM/CDMA
11	Pouze LTE
12	"neznámé"

Pro naše měření jsou používány hodnoty: 9 - pro otestování tzv. user experience, což znamená nastavení připojení mobilního zařízení k nejlepší možné síti. Dále pro otestování GSM sítě nastavíme hodnotu 1, což znamená připojení pouze k síti typu GSM. Pro 3G síť nastavíme hodnotu 2, znamenající připojení pouze přes síť 3G. A pro poslední měření LTE nastavíme hodnotu 11, indikující připojení přes síť LTE. Aby však operační systém Android vzal nová nastavení v potaz a řídil se jimi, je nutné restartovat mobilní telefon. Nebo elegantněji a efektivněji pouze restartovat proces *com.android.phone*. Použijeme tento příkaz:

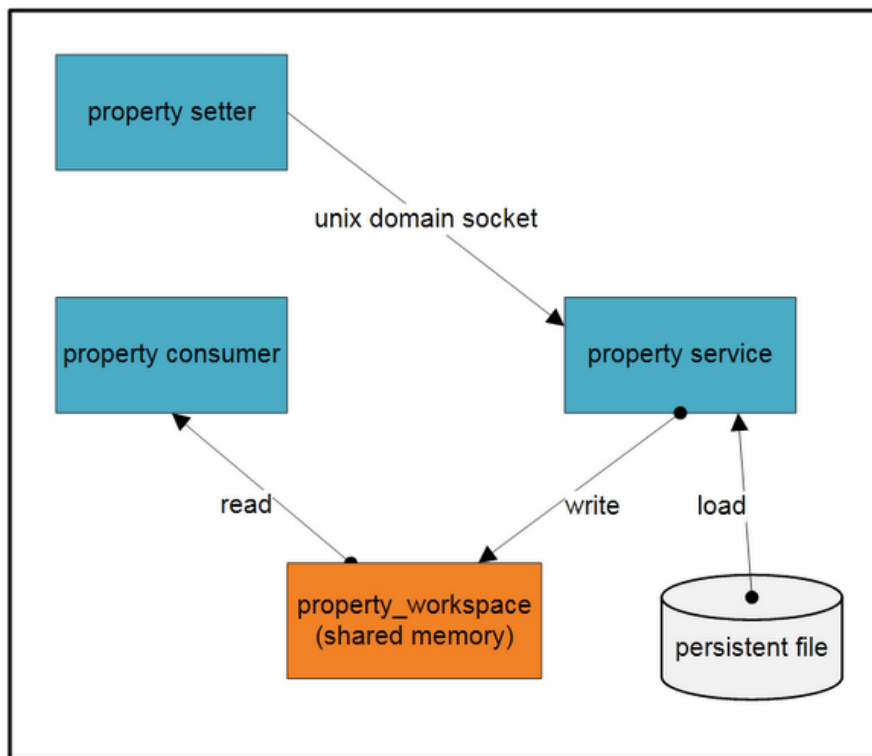
```
killall com.android.phone
```

Po několika málo sekundách jsou načtena nová nastavení a mobilní telefon se připojí k síti, kterou jsme mu předurčili. Alternativou k tomuto poněkud komplikovanějšímu, nicméně plně funkčnímu postupu bylo použití příkazu

```
setprop gsm.network.type typ_sítě
```

Parametr "*typ_sítě*" mohl být například EDGE, LTE, HSPA. Nicméně fungování tohoto příkazu v novějších verzích Androidu není zaručeno. Testoval jsem tento postup na operačních systémech Android verze 4.3 i starší verze 4.2.2 a ani na jednom z testovaných systémů jsem nedosáhl úspěchu. Je nutno počítat se zpožděním mezi učiněním nového nastavení a následným připojením se k novému typu mobilní sítě. Proto je ve skriptu nastaven dostatečný ochranný časový interval mezi zápisem hodnoty nového nastavení, následované připojením se k požadované síti a započítáním samotného měření.

Na počátku každého měřicího cyklu skript nejdříve ověří, zdali je mobilní zařízení připojeno k požadovanému druhu mobilní sítě. To se děje pomocí příkazu *getprop*. Operační systém Android obsahuje celý systém vlastností. Ten běží jako služba a řídí systémovou konfiguraci a stav. Veškerá nastavení a stavy jsou vlastnostmi. Ty jsou uloženy v podobě páru klíč/hodnota, obojí je datového typu string. Svou funkcí jsou podobné registrům používaným v desktopových operačních systémech Windows od firmy Microsoft. Tyto vlastnosti mohou být čteny pomocí příkazu *getprop*, nastavovány pak pomocí příkazu *setprop*. Pokud spustíme příkaz *getprop* bez parametrů, vypíše se nám seznam všech vlastností, které jsou pomocí příkazu dostupné. V případě mobilního telefonu Sony Xperia V s operačním systémem Android verze 4.3 je dostupné celých 334 vlastností. Na mobilním telefonu Samsung Galaxy GT-S5660 s Androidem verze 4.2.2 je k dispozici 289 vlastností. Systém vlastností se skládá ze 3 procesů (*property setter*, *property service* a *property consumer*), skupiny datových souborů s trvalým uložením vlastností (*persistent file*) a sdíleného bloku paměti (*property workspace*). *Property workspace* je kontejnerem pro všechny záznamy vlastností. Do tohoto sdíleného bloku paměti může zapisovat pouze *property service*. Zápis probíhá tak, že hodnoty jsou načteny z *persistent file* a uloženy do *property workspace*. *Property consumer* může číst hodnoty přímo z *property workspace*. *Property setter* pak může nastavovat hodnoty skrze tzv. *unix domain sockets*. Hodnoty, které takto přijdou do *property service*, jsou pak uloženy do *property workspace* i *persistent file*. Vše je znázorněno na následujícím obrázku 5.2.



Obrázek 5.2: Čtení a zápis hodnot do systému vlastností OS Android [9]

Systémové vlastnosti, které jsou načítány při startu operačního systému pomocí tohoto systému vlastností, jsou uloženy v souborech:

- */default.prop*
- */system/build.prop*
- */system/default.prop*
- */data/local.prop*

Zjištění aktuálního typu mobilní sítě, přes které je zařízení připojeno, je možné uskutečnit pomocí příkazu *getprop* následovně:

```
typ_site = $(getprop gsm.network.type)
```

Hodnoty, kterých takto definovaná proměnná "*typ_site*" může nabývat, jsou pro naše účely především GPRS, EDGE, HSPA, HSDPA, HSPAP, HSUPA, UMTS a LTE. Díky tomu skript ověří, zdali jsme úspěšně připojeni přes požadovaný typ mobilní sítě a spustí měření, případně měřicí cyklus pro daný typ sítě z důvodu její nedostupnosti nevykoná. V případě úspěchu, tj. úspěšného připojení mobilního telefonu na požadovaný typ sítě, je volán skript *laccellid.sh*, který má za úkol pomocí AT příkazů vytvořit pomocný textový soubor a do něj vložit aktuální hodnoty LAC (Location Area Code) a Cell ID. Tyto dva parametry slouží k upřesnění měření, kde bylo prováděno a vůči které základnové stanici. Podrobnější informace

k tomu, jak skript *laccellid.sh* získává údaje LAC a Cell ID, jsou k dispozici v kapitole 5.3. Výsledkem je pomocný soubor *pom.txt*, ze kterého kombinací příkazů *cat*, *grep* a *awk* dostaneme požadované informace. Například pro získání hodnoty LAC použijeme příkaz:

```
LAC=$(cat pom.txt | grep +CREG: | awk -F' "' '{ print $2 }')
```

Obdobně pak postupujeme v případě parametru Cell ID.

5.2.2 Výpočet přenosové rychlosti

Stanovení nejdůležitějšího údaje našeho měření - rychlosti stahování dat (download) - je realizováno pomocí nástroje *wget*. Jeho použití je podobné jako v případě využívání tohoto nástroje v prostředí operačního systému typu Linux.

[10] Wget je volně šiřitelný softwarový nástroj, který slouží ke stahování souborů skrze HTTP, HTTPS a FTP - skrze nejpoužívanější Internetové protokoly. Jedná se o neinteraktivní nástroj příkazové řádky, hodí se proto pro použití v rámci skriptů, nebo může být volán pomocí *cronu*. Umožňuje navázat u přerušených stahování, vytvářet obrazy celých webů nebo FTP serverů, je multiplatformní a má mnoho dalších výhod a předpokladů, díky kterým se hodí pro stanovení rychlosti připojení ve směru download.

Výsledný příkaz pro stahování souboru ze vzdáleného serveru a stanovení tak rychlosti stahování dat v mobilní síti je:

```
nohup wget -O /dev/null http://speedtest.cambium.nl/100mb.bin &
```

V tomto případě je použit příkaz *nohup*, který způsobí, že stahování souboru přes *wget* běží na pozadí a další činnost skriptu tak není narušena ani omezena. Syntaxe příkazu *nohup* je následující:

```
nohup jakýkoliv_prikaz_ktery_ma_bezet_na_pozadi &
```

Takto se spustí "jakýkoliv příkaz, který má běžet na pozadí" a vykonávání skriptu pokračuje dál, nehledě na to, jaký příkaz a jak dlouho běží na pozadí. To nám umožní specifikovat dobu, po jakou má běžet stahování souboru a po této době stahování přerušit.

Nástroji *wget* pomocí parametru *-O* je stanoveno, aby ukládal stahovaný soubor do */dev/null*. */dev/null* se tváří jako hardwarové zařízení, jedná se však pouze o softwarovou emulaci hardwarového zařízení v systému. Při čtení je obsah */dev/null* prázdný a při zápisu jsou všechna data skartována. Posílání stažených dat do */dev/null* je zvoleno proto, aby u některých zařízení nedocházelo k omezení maximální přenosové rychlosti při měření vyplývající z limitující rychlosti ukládání dat na paměťovou kartu. Tento problém se týká především stahování prostřednictvím sítě LTE, kde mohou být přenosové rychlosti v řádech až stovek megabitů za sekundu.

Testování přenosové rychlosti probíhalo přes několik různých testovacích serverů. Nakonec se jako nejpoužitelnější ukázal testovací server na adrese <http://speedtest.cambium.nl/>.

Výhodný je například z toho pohledu, že z něj můžeme stahovat jakkoliv velký datový soubor. Záleží jen na nás, jak velký soubor potřebujeme stáhnout. Testovací server nás v tomto nijak neomezuje. Není však problém nastavit jakýkoliv jiný testovací server, případně si vytvořit svůj vlastní, který však musí být vybaven dostatečně rychlou konektivitou, aby nedocházelo ke zkreslení měření.

Pro stanovení přenosové rychlosti ve směru upload, čili od mobilního zařízení k síti Internetu, je použit nástroj *ftpput*. Jedná se o nástroj spouštěný prostřednictvím příkazové řádky, který odesílá soubory z lokálního úložiště (paměť mobilního telefonu) na vzdálený FTP (File Transfer Protocol) server. K testovacím účelům bylo vyzkoušeno několik FTP serverů, při tom nebyl nalezen žádný problém ani s jedním z nich. Vždy bylo použito bezproblémové, stačí pouze kapacita takového FTP serveru alespoň 100 MB a dostatečná konektivita umožňující nahrávání souborů rychlostí až několika desítek megabitů za sekundu. Výsledný *ftpput* příkaz je následující:

```
ftpput -v --user=uživatel --pass=heslo ftp_server.cz data1 data2
```

Výsledkem příkazu je, že dojde k nahrání souboru "*data1*" na vzdálený FTP server s adresou "*ftp_server.cz*". K přihlášení na FTP server je použit uživatelský účet "*uživatel*" a heslo "*heslo*". Zdrojový soubor "*data1*", který je uložen v lokální paměti mobilního telefonu, je ukládán na vzdálený FTP server pod názvem "*data2*". Po uplynutí doby měření už pro nás není další přenos souboru relevantní, je proto ukončen a data nahraná na FTP server jsou smazána, aby byl uvolněn dostatek místa pro další měření. V tomto případě jsou hodnoty uživatelského účtu, hesla a vzdáleného FTP serveru pouze demonstrační, v praxi se musí nahradit existujícími reálnými údaji.

Stanovení množství přenesených dat během měření probíhá pomocí informací obsažených ve struktuře */proc/* souborového systému. Jedná se určitým způsobem o trik, jakým jádro operačního systému Linux, potažmo Android, zprostředkovává určité interní informace dostupné ostatním uživatelským procesům a aplikacím. Jádro systému tyto informace prezentuje ve formě virtuálních souborů a složek, které jsou hierarchicky uloženy ve složce */proc/*. Soubory a složky */proc/* souborového systému jsou virtuální, jelikož nejsou uloženy v žádném druhu permanentní paměti, jako je pevný disk nebo paměťová karta. Namísto toho jsou veškerá data zde uložená vytvářena dynamicky přímo jádrem operačního systému a kdykoliv tak dostupná ke čtení ostatním procesům. Pro nás důležité údaje jsou dostupné v */proc/net/*. Zde jsou k dispozici různé údaje týkající se sítě. Konkrétně v */proc/net/dev/* jsou obsaženy údaje o množství přenesených dat skrze různé typy datových sítí. Struktura obsahuje následující sloupce (údaje): *Interface | Received bytes; packets; errors; drop; fifo; frame; compressed; multicast | Transmitted bytes; packets; errors; drop; fifo; colls; carrier; compressed*. Řádky struktury pak tvoří následující rozhraní: *lo; dummy0; rmnet0; rmnet1; rmnet2; rmnet3; rmnet4; rmnet5; rmnet6; rmnet7; tunl0; gre0; sit0; ip6tln0*. Nejdůležitější údaje jsou na řádce *rmnet0*, obsahují totiž informace týkající se mobilní datové sítě, kterou testujeme. Především pak sloupce "*Received bytes*" a "*Transmitted bytes*". Tyto hodnoty jsou zaznamenány vždy na

začátku měření a na jeho konci. Pro zaznamenání stavu přenesených bajtů v rámci mobilní datové sítě slouží tato kombinace příkazů:

```
stavdownloadu=$(cat /proc/net/dev | grep rmnet0: | awk '$2>0  
{print $2}')
```

```
stavuploadu=$(cat /proc/net/dev | grep rmnet0: | awk '$10>0  
{print $10}')
```

Příkazy pro zjištění množství přenesených dat ve směru download i upload jsou téměř totožné, liší se pouze pořadím sloupce, který potřebujeme zjistit. Ve směru download se jedná o 2. sloupec, ve směru upload jde o 10. sloupec. Pomocí nástroje *cat* dojde k výpisu textových informací uložených ve struktuře */proc/net/dev/*, pomocí příkazu *grep* se vypíše pouze pro nás zajímavý řádek s informacemi o rozhraní "*rmnet0*" a pomocí nástroje *awk* se vypíše pouze ta hodnota, kterou potřebujeme znát. Čili pouze množství přenesených bajtů. Výsledná hodnota přenosové rychlosti má být v kilobitech za sekundu, proto je nutné nejdříve veškeré nasbírané hodnoty převést právě z bajtů na kilobity. Poté již stanovím výslednou hodnotu přenosové rychlosti pomocí vztahu:

$$v = \frac{d_2 - d_1}{t} \quad [\text{kbit/s}; \text{kbit}; \text{kbit}; \text{s}]$$

v ... přenosová rychlost; d₂ ... počet kilobitů po měření;

d₁ ... počet kilobitů před měřením; t ... doba měření

Rovnice (5.1)

Výsledná přenosová rychlost je tedy stanovena jako rozdíl množství přenesených dat po měření a množství přenesených dat při započetí procesu měření. Tento rozdíl je vydělen délkou měření v sekundách. Výsledkem je hodnota přenosové rychlosti v kilobitech za sekundu. Stejným způsobem je stanovena jak přenosová rychlost ve směru od mobilního zařízení k síti, tak ve směru od mobilní sítě k zařízení.

5.2.3 Způsob měření podle technické zprávy ETSI TR 102 678

Použitý způsob měření odpovídá technické zprávě ETSI TR 102 678. ETSI je Evropský ústav pro telekomunikační normy. Jedná se o nezávislou a neziskovou organizaci zabývající se standardizací v telekomunikačním průmyslu. Ačkoliv se jedná o evropskou organizaci, její rozhodnutí mají celosvětový dopad.

Dle doporučení vyplývajících z technické zprávy ETSI TR 102 678 má měření probíhat pomocí stahování a odesílání datového souboru. Přijímaný a odesílaný datový soubor musí mít dostatečnou velikost k tomu, aby nemohlo dojít k situaci, že bude celý odeslán nebo stažen ještě před uplynutím pevně stanovené doby měření. Během vývoje měřicí sondy se ukázala doba 10 sekund jako dostatečná pro měření přenosové rychlosti jak pro stahování, tak i pro odesílání

dat, čili ve směru od mobilního telefonu k síti i naopak. Měření rychlosti stahování probíhá pomocí nástroje *wget*, kdy dochází ke stahování datového souboru umístěného na vzdáleném serveru. Tento soubor je dostatečně velký na to, aby ani teoreticky nemohl být při maximálně možných dosažitelných rychlostech mobilních datových sítí stažen během doby měření, tj. 10 sekund. Datový soubor, jak již bylo zmíněno, je ukládán do */dev/null/*, což má za následek, že stažená data jsou automaticky zahazována (skartována) a nejsme tak limitováni maximální možnou rychlostí zápisu dat do interní paměti mobilního zařízení či na paměťovou kartu. Obdobným způsobem probíhá i stanovení rychlosti odesílání dat. Dostatečně velký datový soubor je kvůli vyšším rychlostem čtení uložen v interní paměti mobilního zařízení, která je pro potřeby měření dostatečně rychlá - rychlejší než při čtení z paměťové karty. Tento soubor je pomocí nástroje *ftpput* odesílán na vzdálený FTP server.

Podle přesné specifikace technické zprávy ETSI TR 102 678 musí být přenos řádně ukončen, až poté může následovat výpočet a stanovení množství přenesených dat, potažmo přenosové rychlosti. Rozpad spojení je pak iniciován měřicí sondou, která spojení předčasně po uplynutí předem pevně stanovené doby měření ukončí.

Technická zpráva ETSI TR 102 678 je k dispozici v seznamu použité literatury [11].

5.2.4 Stanovení délky zpoždění (latence)

Stanovení délky zpoždění je uskutečněno pomocí nástroje Ping. Ping využívá ICMP protokol. Funguje tak, že odesílá na vzdálený server tzv. *ECHO_REQUEST* datagram a čeká od něj *ECHO_RESPONSE* odpověď. Podle délky prodlevy mezi odesláním *ECHO_REQUEST* datagramu a přijetím *ECHO_RESPONSE* datagramu je stanovena konkrétní doba zpoždění, tzv. *latence*. Konkrétní příkaz použitý v rámci testovací sondy je následující:

```
odezva=$(ping -c 20 www.google.com | tail -1 | awk '{print $4}'  
| cut -d '/' -f 2)
```

Příkaz *ping* je pomocí přepínače *"-c 20"* spuštěn 20 krát, aby bylo pořízeno dostatečné množství měření. Pomocí příkazu *"tail -1"* je vyfiltrován pouze poslední řádek z výpisu obsahující výsledné hodnoty měření. Pomocí příkazů *"awk '{print \$4}'"* a *"cut -d '/' -f 2"* je docíleno toho, že do proměnné *"odezva"* je uložena výsledná průměrná hodnota zpoždění z uplynulých dvaceti vzorků měření. Stanovení doby odezvy je realizováno vůči serveru společnosti Google.

5.2.5 Zjištění dalších parametrů měření

Další doplňkové informace, které jsou součástí výsledků měření, jsou získávány především prostřednictvím nástroje *dumpsys*. Tento nástroj běží na mobilním zařízení a poskytuje informace o stavu systémových služeb. V příkazové řádce stačí napsat příkaz:

```
dumpsys
```

Výsledkem je výpis všech běžících služeb, o kterých nástroj *dumpsys* poskytuje informace. Týká se to především veškerých nastavení v systému (účty, budíky, zvuky, bluetooth, baterie, kamera, přístupová práva, nastavení sítě, apod.). V případě mobilního zařízení Sony Xperia V s operačním systémem Android verze 4.2 se jedná celkem o 36 604 různých hodnot nastavení. Použití nástroje je následující:

```
dumpsys [název_běžící_služby]
```

Název běžící služby může být například "*telephony.registry*". Výsledkem může být následující výpis:

```
Last known state:
mCallState=0
mCallIncomingNumber=
mServiceState=0 home T-Mobile CZ T-Mobile CZ 23001 (Manual) EDGE
CSS not supported -1 -1 RoamInd=-1 DefRoamInd=-1 EmergOnly=false
mSignalStrength=SignalStrength: 11 -1 -120 -160 -120 -1 -1 99 -1
-1 -1 -1 gsm|lte
mMessageWaiting=false
mCallForwarding=false
mDataActivity=0
mDataConnectionState=0
mDataConnectionPossible=true
mDataConnectionReason=2GVoiceCallEnded
mDataConnectionApn=
mDataConnectionLinkProperties=null
mDataConnectionLinkCapabilities=null
mCellLocation=Bundle[mParcelledData.dataSize=76]
mCellInfo=null
registrations: count=8
```

Nejdůležitější z předchozího výpisu pomocí nástroje *dumpsys* je řádek "*mSignalStrength*". Obsahuje totiž informace týkající se aktuální úrovně přijímaného signálu ze základnové stanice. Doplnkové parametry, které jsou v tomto řádku uvedeny, jsou následující:

Tabulka 5.2: *Tabulka parametrů v řádku SignalStrength [12]*

Parametr	Jednotka	Význam
mGsmSignalStrength	[-]	Síla signálu (hodnoty 0-31); přepočet na [dBm] je: $((mGsmSignalStrength*2)-113)$
mGsmBitErrorRate	[-]	Chybovost (hodnoty 0-7)
mCdmaDbm	[dBm]	CDMA RSSI - indikátor síly přijímaného signálu
mCdmaEcio	[dB*10]	CDMA Ec/Io - poměr přijaté energie a energie šumu
mEvdoDbm	[dBm]	EVDO RSSI - indikátor síly přijímaného signálu
mEvdoEcio	[dB*10]	EVDO Ec/Io - poměr přijaté energie a energie šumu
mEvdoSnr	[-]	EVDO SNR - poměr síly signálu k síle šumu (hodnoty 0-8)
mLteSignalStrength	[-]	Síla signálu (hodnoty 0-31); přepočet na [dBm] je: $((mLteSignalStrength*2)-113)$
mLteRsrp	[dBm]	Síla referenčního (pilotního) signálu buňky
mLteRsrq	[dB]	Kvalita referenčního signálu (jedná se o poměr RSSI a RSRP)
mLteRssnr	[dB]	LTE SNR - poměr síly signálu k síle šumu
mLteCqi	[-]	Kvalita kanálu
Network_type	[-]	Typ sítě - GSM, CDMA, LTE

Ne všechny hodnoty, které jsou uvedeny v tabulce 5.2, jsou pro naše měření relevantní. Nejdříve se před zjišťováním doplňkových parametrů ve skriptu ověří, zdali se jedná o připojení typu GSM, 3G nebo LTE a podle toho se do výsledků zapíše pouze vybrané údaje. V případě LTE sítě se zapíše údaje o síle signálu, RSRP, RSRQ, SNR. V ostatních případech se zaznamenají do výsledků údaje o síle signálu a chybovosti BER. Ve všech případech jsou navíc zaznamenány u všech měření údaje o rychlosti stahování dat, odesílání dat, odezvě, typu mobilní datové sítě, LAC a Cell ID. Každý záznam měření má navíc časové razítko obsahující údaje o datu a času měření a také v rámci sítě jakého operátora bylo měření prováděno.

5.2.6 Odeslání výsledků

Po proměření nejdříve defaultního typu sítě po zapnutí mobilního telefonu, následně 2G, 3G a LTE sítě jsou veškeré výsledky uloženy do tzv. CSV (Comma-separated values) souboru. Čili do jednoduchého textového souboru, jehož jednotlivé položky jsou odděleny středníkem a slouží k výměně tabulkových dat. Tento výsledný soubor je následně odeslán pomocí nástroje "*fipput*" na vzdálený FTP server k dalšímu zpracování. Výsledky měření mohou být následně importovány do databáze a poté například vizualizovány pomocí grafů.

5.3 Doplnkový skript *laccellid.sh*

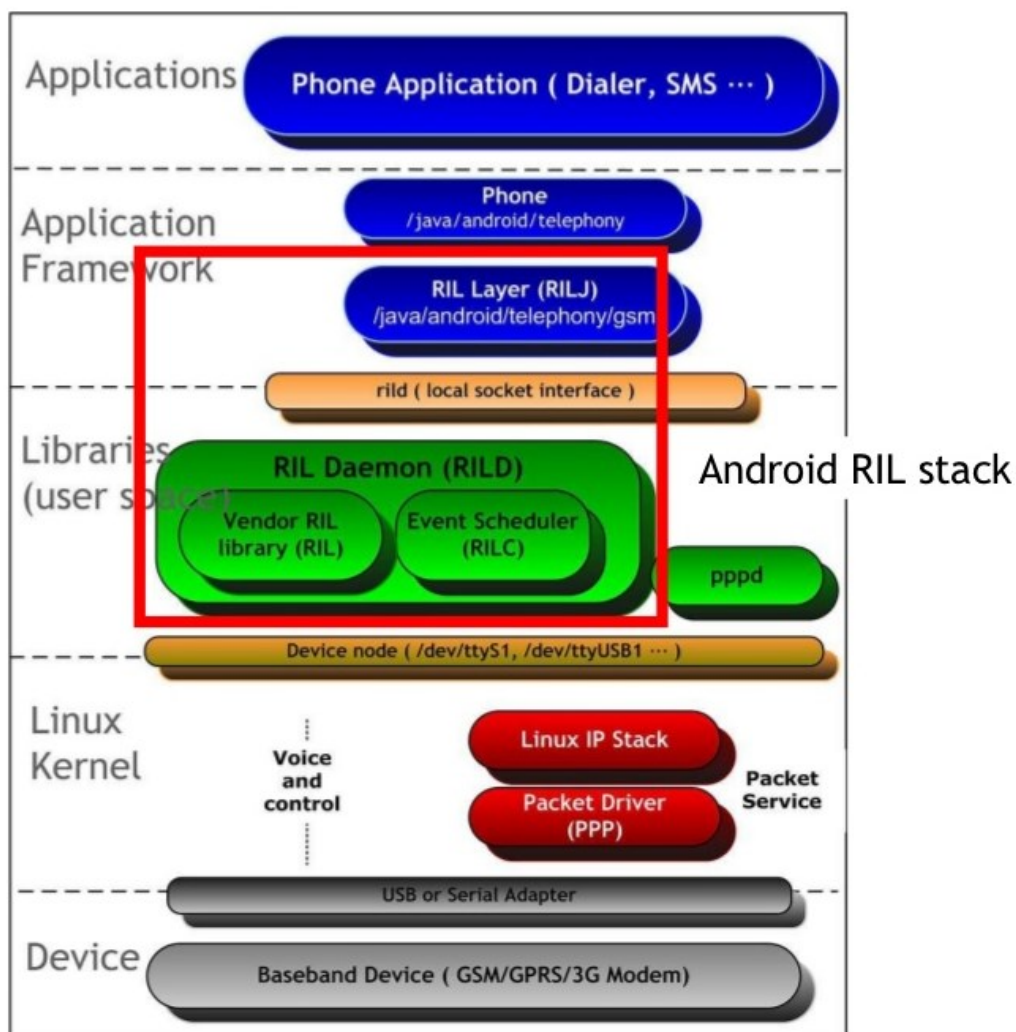
Doplnkový skript *laccellid.sh* je ideálně uložen ve stejné složce, jako hlavní měřicí skript *test.sh*. V našem případě ve složce */sdcard/*. Není to však nezbytná podmínka. Tento skript je volán a využíván skriptem *test.sh*. Po splnění svého úkolu je předáno řízení měření opět hlavnímu skriptu *test.sh*. Skript slouží ke zjištění údajů LAC (Location Area Code) a Cell ID (Cell Identification). Tyto dva údaje slouží k bližší lokalizaci a určení místa měření a použité základnové stanice BTS.

LAC (Location Area Code) je kód určité oblasti, území, které pokrývají signálem desítky až stovky základnových stanic. Tyto základnové stanice jsou seskupeny dohromady nejen kvůli optimalizaci signalizace. Obvykle jsou základnové stanice v rámci jedné oblasti řízeny stejným BSC (Base Station Controller) v GSM sítích či stejným RNC (Radio Network Controller) v sítích UMTS. Právě díky sdílení řídicích BSC či RNC v mobilních sítích je umožněn handover, lepší optimalizace a využití sítě a podobně. K identifikaci toho, v jaké lokální oblasti se právě nachází mobilní zařízení, používá se již zmiňované LAC. To je vysíláno v pravidelných časových intervalech základnovými stanicemi. LAC se skládá z maximálně 5 decimálních čísel nebo 8 dvoj bitů kódovaných v hexadecimálním tvaru. Platí, že $LAC < FFFF$.

Cell ID je jednoznačné a unikátní identifikační číslo definované mobilním operátorem a přiřazené každé buňce v mobilní síti. Ve 2G sítích má velikost 16 bitů, ve 3G sítích má velikost 32 bitů. Nabývá tedy hodnot 0 až 65 535 v GSM sítích, v UMTS a LTE sítích 0 až 268 435 455 při zápisu v decimálním tvaru.

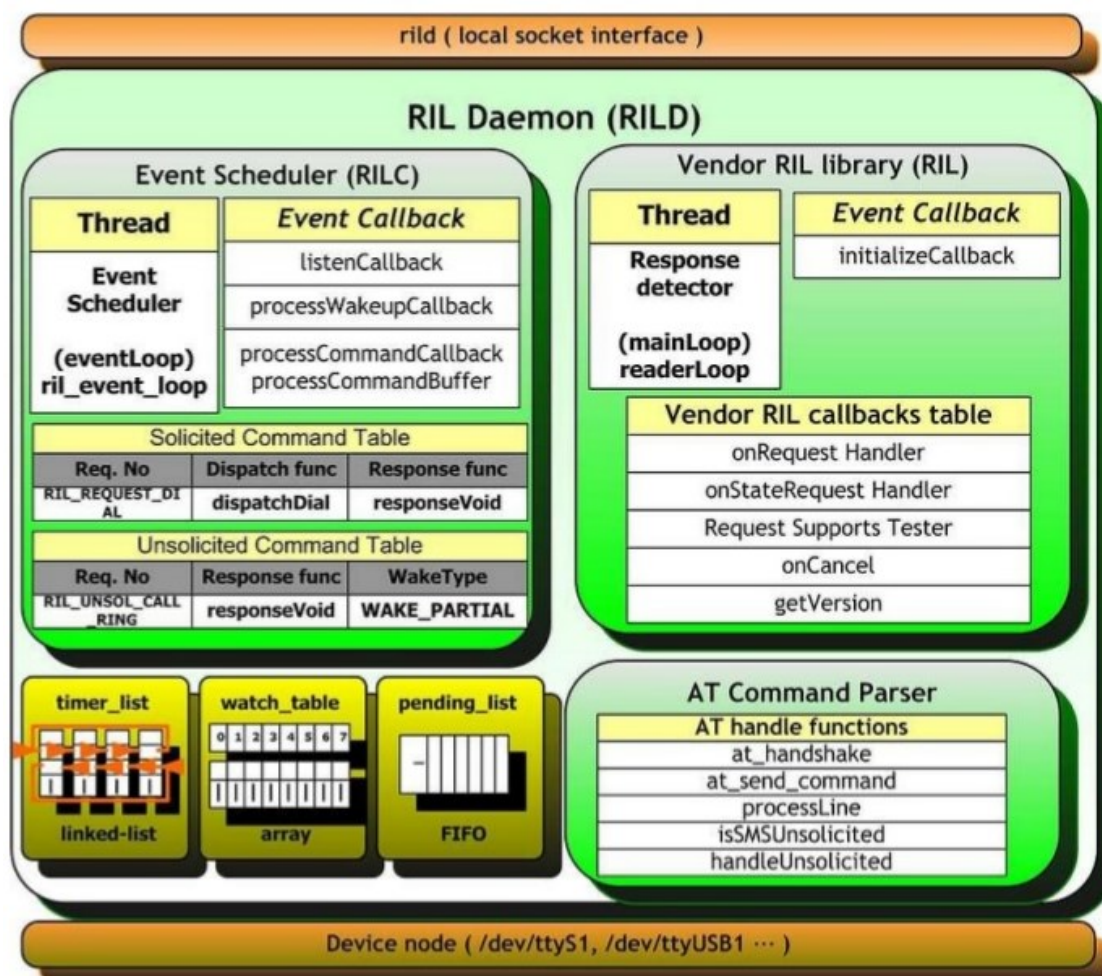
Ke zjištění hodnot LAC a Cell ID nelze použít standartní příkazy, jako je *getprop* nebo *dumpsys*. Bohužel tyto údaje nejsou standartní cestou v Androidu dosažitelné, není totiž možné komunikovat napřímo s rádiovým modemem obsaženým v mobilním telefonu. Rádiový modem je v podstatě zařízení typu modulátor-demodulátor sloužící k přenosu digitálního signálu skrze telefonní síť. Abstraktní vrstvu mezi Androidem (*android.telephony*) a rádiovým hardwarem představuje tzv. RIL (*Radio Interface Layer*). RIL se skládá ze dvou základních částí: *RIL Daemon* a *Vendor RIL*. *RIL Daemon* slouží k inicializaci *Vendor RIL*, zpracovává veškerou komunikaci z operačního systému Android směřující na rádiový modem a odesílá volání vrstvě *Vendor RIL* jako vyžádané příkazy. *Vendor RIL* zpracovává komunikaci s rádiovým modemem

a odesílá volání démonu *rild* (RIL Daemon) skrze nevyžádané příkazy. Schéma znázorňující fungování a strukturu RIL je k dispozici na obrázku 5.3.



Obrázek 5.3: Znáznornění vrstvy RIL [13]

Pro používání funkcí rádiového hardwaru a zpřístupnění tak údajů LAC a Cell ID jsou použity tzv. *AT příkazy* (Attention Commands). Jedná se o průmyslový standard, původně vyvinutý firmou Hayes k ovládání modemů, poprvé byly publikovány již v roce 1981. Vyvinuty byly pro jednodušší konfiguraci modemu. Zmiňované AT příkazy jsou posílány skrze démona RILD. Vnitřní struktura RILD démonu je k dispozici na obrázku 5.4.



Obrázek 5.4: Vnitřní struktura démona RILD [13]

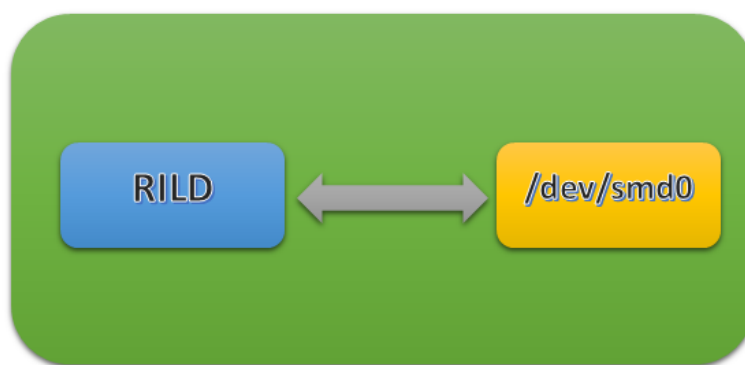
Používání AT příkazů vyžaduje mít *root* přístup k telefonu. Pro otestování, zdali je používání AT příkazů k dispozici, můžeme zadat přes ADB shell běžící na počítači, ke kterému je připojen mobilní telefon, nebo přímo v mobilním telefonu přes Terminal Emulator příkaz:

```
echo "AT\r" > /dev/smd0
```

Tím dojde k odeslání testovacího AT příkazu. Odpověď může být *OK* nebo *ERROR*. Abychom si však mohli přečíst odpověď na uvedený AT příkaz, musíme do terminálu zadat následující:

```
cat /dev/smd0
```

Soubor */dev/smd0* je v podstatě pseudo-terminálový soubor, který využívá RIL démon, potažmo operační systém Android, ke komunikaci s rádiovým modemem. RILD otevírá */dev/smd0* jako slave konec pseudo-terminálu a používá jej jako vstupně/výstupní zařízení k vydávání příkazů a k obdržení odpovědí. V praxi to vypadá jako na obrázku 5.5.



Obrázek 5.5: Komunikace mezi RILD a /dev/smd0

Pokud se nám dostane odpovědi OK, nic nám nebrání v používání AT příkazů na daném zařízení. Žádná odpověď pak indikuje nějakou závadu. Na vině může být například využívání pseudo-terminálu pro komunikaci s rádiovým modemem na jiné adrese, než je /dev/smd0. Některá zařízení využívají /dev/smd11 nebo /dev/ttyGS0.

Každý výrobce mobilních zařízení může mít podporu pro různé AT příkazy, některé nemusí být podporovány vůbec, jiné zase mohou být speciálně vyvinuté pro zařízení konkrétních výrobců. Z toho důvodu je důležité znát příkaz:

```
echo "AT+CLAC\r" > /dev/smd0
```

Jako odpověď nám přijde seznam všech AT příkazů, které jsou podporovány konkrétním zařízením. Většinou se jedná až o několik desítek příkazů. Jeden ze základních a poměrně zajímavých příkazů je:

```
echo "ATI\r" > /dev/smd0
```

Odpovědi jsou nám informace o výrobcu zařízení, modelu, revizi a IMEI. Pro nás však nejdůležitější jsou následující dva příkazy:

```
echo "AT+CREG=2" > /dev/smd0
```

```
echo "AT+CREG?" > /dev/smd0
```

První příkaz řekne rádiovému modem, že má povoleno odesílat informace o síti a poloze skrze CREG. Toho využijeme hned v následujícím příkazu AT+CREG?, jehož odpověď je například následující:

```
+CREG: 1, 1, 7032, AA66
```

První parametr znamená status registrace mobilního zařízení v síti, v tomto případě značí, že je mobilní zařízení registrováno v domácí síti, druhý parametr značí typ přístupové sítě, u nás se jedná o GSM síť, 7032 je LAC (Local Area Code) v hexadecimálním tvaru a AA66 je Cell ID v hexadecimálním tvaru. Možné hodnoty, kterých může nabývat první parametr v odpovědi CREG, jsou v tabulce 5.3. Možné hodnoty druhého parametru pak v tabulce 5.4.

Tabulka 5.3: *Hodnoty statusu registrace mobilního zařízení v síti (1. parametr CREG) [14]*

Hodnota	Význam
0	Neregistrováno, nevyhledává nového operátora
1	Registrováno v domácí síti
2	Neregistrováno, vyhledává nového operátora
3	Registrace zakázána
4	Neznámé
5	Registrováno v cizí síti (roaming)
6	Registrováno "pouze SMS" v domácí síti
7	Registrováno "pouze SMS" v cizí síti
8	Pouze tísňová volání
9	Nemožnost hovoru v domácí LTE síti
10	Nemožnost hovoru v cizí LTE síti

Tabulka 5.4: *Hodnoty technologie přístupové sítě (2. parametr CREG) [14]*

Hodnota	Význam
0	GSM
1	GSM Compact
2	UTRAN
3	GSM/EGPRS
4	UTRAN/HSDPA
5	UTRAN/HSUPA
6	UTRAN/HSDPA a HSUPA
7	E-UTRAN

5.4 Funkce watchdog

Nedílnou součástí řešení diplomové práce je funkce *watchdog*, tzv. hlídací pes. Jedná se o mechanismus, který nepřímo souvisí s výkonem hlavní části diplomové práce - testováním mobilních datových sítí. Nicméně měřicí sondu doplňuje a rozšiřuje možnosti jejího použití především v různých odlehlých místech, kde není možnost každodenně kontrolovat, zdali sonda správně funguje. Testovací sonda funguje díky naprogramovaným skriptům a správnému nastavení naprosto samostatně, nevyžaduje žádnou lidskou interakci, nicméně se v praxi občas stává, že mobilní zařízení, v tomto případě mobilní telefon s operačním systémem Android občas „zamrzne“ nebo se začne chovat nepředvídatelně, nesprávně.

Abych předešel možnému neočekávanému chování mobilního telefonu, který bude umístěn na místech vzdálených od obsluhy, která by jej mohla restartovat, navrhl jsem pomocí platformy Arduino funkci pro pravidelné restartování mobilního zařízení. Díky pravidelnému restartování mobilního telefonu můžeme buďto úplně zamezit neočekávanému chování a zvláštním výsledkům měření, případně můžeme omezit a minimalizovat výskyt tohoto chování, například „zamrznutí“ mobilního telefonu a úplnému výpadku testování mobilních sítí. Pokud by se však už takové chování objevilo, bylo by nejpozději s dalším restartováním mobilního telefonu odstraněno a testování by probíhalo dále. Nastavení intervalu pro pravidelný restart mobilního zařízení je věc individuální. Můžeme nastavit pravidelný restart každých 24 hodin, nebo můžeme zvolit například restartování pouze jedenkrát týdně. Tento časový interval musíme zvolit s ohledem na dobu mezi jednotlivými měřeními přenosových parametrů mobilních sítí pomocí testovací sondy. Pokud by bylo například zvoleno měření přenosových parametrů pomocí testovací sondy dvakrát denně a zároveň by bylo nastaveno pravidelné restartování měřicí sondy pomocí platformy Arduino každých 24 hodin, v případě „zamrznutí“ či jiné chyby v testovací sondě bychom tak měli maximálně dvě chybná, neúplná, nebo kompletně neprovedená měření. Po dalším restartování se by však testovací sonda opět fungovala normálně a opět prováděla automatizovaně veškerá další měření. Mnou navržená funkce hlídacího psa sestává ze dvou částí. První částí je testovací sonda, druhou částí je zařízení zkonstruované pomocí platformy Arduino.

Aby došlo k pravidelnému restartování mobilního telefonu, vymyslel jsem následující postup. Přišel jsem totiž na fakt, že když máme mobilní telefon s operačním systémem Android neustále připojen na nabíječku a pokud mu během provozu vyjmeme baterii, dojde k jeho vypnutí. Pokud je operační systém Android na mobilním telefonu správně přizpůsoben, po opětovném vložení baterie dojde k samovolnému a automatickému zapnutí mobilního telefonu. Operační systém telefonu naběhne do provozuschopného stavu, jako bychom telefon zapnuli ručně. Veškerá měření tak mohou dál pokračovat. Jsou totiž realizována pomocí skriptů, které jsou automaticky spouštěny při startu mobilního telefonu.

5.4.1 Úprava operačního systému Android v měřicí sondě

Mobilní telefon, kterému je i přes připojenou nabíječku vytažena baterie během provozu, se automaticky vypne. Platí to v drtivé většině případů. Nicméně jak docílit toho, aby se při opětovném vložení baterie zpět znovu nastartoval? Možná řešení, která jsem našel a vyzkoušel jejich funkčnost, jsou dvojí. Oba postupy vyžadují mobilní telefon s tzv. *root* přístupem, čili s právy administrátora.

První řešení spočívá v tom, že musíme nejdříve najít tzv. *battery animation file*. Jedná se o soubor, který má na starosti spuštění animace ilustrující nabíjení baterie mobilního telefonu, který je vypnutý ale zároveň nabíjí baterii pomocí nabíječky. Umístění a název souboru je různý podle výrobce mobilního zařízení a podle verze operačního systému. Obvykle se jedná o jedno z těchto umístění a názvu souboru:

```
/system/bin/chargemon  
/system/bin/lpm  
/system/bin/playlpm  
/system/bin/pm  
/system/bin/ipod
```

Obsah tohoto souboru stačí nahradit následující posloupností příkazů:

```
#!/system/bin/sh  
su  
/system/bin/reboot
```

V praxi se mi dokonce povedlo docílit stejného efektu i pouhým přejmenováním onoho souboru zodpovědného za zobrazení ilustrace nabíjení baterie či jeho úplným vymazáním. Pomocí dlouhodobého testování tohoto postupu je ověřeno, že smazání či editace tohoto souboru nemá žádné jiné vedlejší efekty, nepředstavuje žádné riziko pro mobilní zařízení.

Druhé řešení pak spočívá ve využití tzv. režimu *fastboot*. Jedná se o speciální diagnostický a inženýrský protokol, do kterého lze zapnout mobilní zařízení s operačním systémem Android. Tento režim však není defaultně pro uživatele dostupný u všech zařízení s OS Android. Nicméně jde o mocný nástroj, pomocí kterého můžeme upravovat přímo obraz operačního systému mobilního telefonu pomocí počítače skrze USB připojení. Do *fastboot* režimu mobilního zařízení můžeme přistupovat z operačního systému Windows, Linux nebo MAC. Musíme však mít speciální USB ovladače pro konkrétní typ mobilního telefonu. V případě mobilního telefonu Sony Xperia V, kde je *fastboot* režim standardně uživatelům přístupný, se do něj dostaneme následujícím způsobem: Nejdříve vypneme mobilní telefon. Poté stiskneme a držíme hardwarové tlačítko pro zvýšení hlasitosti. Během stisknutého tlačítka pro zvýšení hlasitosti připojíme mobilní telefon pomocí USB kabelu k počítači a tlačítko pustíme. Telefon by měl automaticky najet do *fastboot* režimu. O tom, zdali vše proběhlo korektně a mobilní telefon opravdu najel do *fastboot* režimu, se můžeme přesvědčit pomocí toho, že

informační dioda mobilního telefonu změnila svou barvu na modrou. Do *fastboot* režimu se můžeme alternativně dostat také pomocí ADB nebo speciální aplikace třetí strany stažené z Google Play, například aplikace s názvem *Quick Boot (Reboot)* je k dispozici ke stažení zdarma v obchodě Google Play. Nicméně mnou popsáný způsob je nejjednodušší.

Jakmile se dostaneme do *fastboot* režimu, skrze konzoli počítače pošleme do mobilního telefonu následující sekvenci příkazů:

```
fastboot oem unlock  
fastboot oem off-mode-charge 0  
fastboot oem lock  
fastboot reboot
```

První příkaz má za následek odemčení záložního obrazu operačního systému, abychom v něm mohli provádět úpravy. Druhý příkaz pak nastaví zařízení do stavu, aby se automaticky zapnulo při vložení baterie ve chvíli, kdy je navíc ještě k mobilnímu telefonu připojena nabíječka. Třetí příkaz opět zamkne záložní obraz operačního systému, aby v něm už nemohly být prováděny jakékoliv změny. Pomocí posledního příkazu restartujeme mobilní telefon s již novým nastavením.

Tyto postupy jsem vyzkoušel v praxi na mobilním telefonu Sony Xperia V s operačním systémem Android verze 4.3, HTC Wildfire s OS Android 2.3.7 a na Samsungu GT-S5660 s operačním systémem Android verze 4.2.2. Ve všech případech jsem byl úspěšný a byla tímto dokázána funkčnost tohoto řešení. Jedinou výjimku tvoří mobilní telefon Phicomm C230w se systémem Android ve verzi 4.3. Tento telefon, který pochází z Číny, se totiž chová odlišně od ostatních běžných telefonů. Zjistil jsem, že když je tento telefon připojen na nabíječku a během provozu mu vytáhneme baterii, telefon normálně běží dál, jako by se nic nedělo. V ostatních případech u telefonů jiných výrobců dojde k automatickému vypnutí mobilního telefonu a s telefonem až do opětovného vložení baterie nelze nic dělat. U tohoto telefonu bychom tedy náš postup nemohli realizovat.

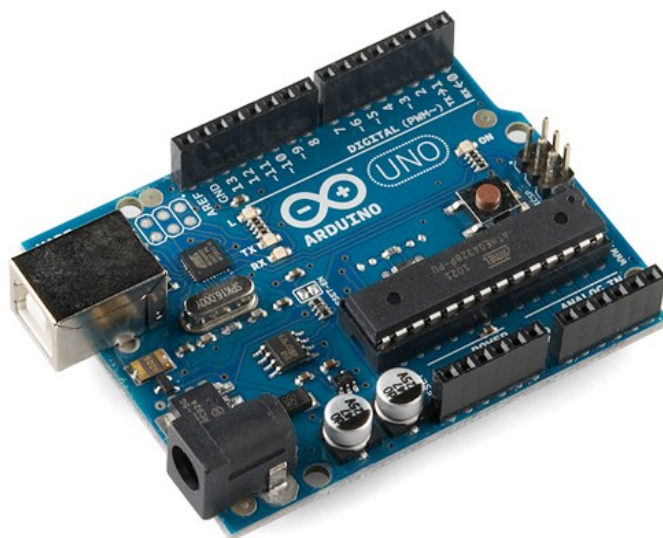
Pokud jsme úspěšně aplikovali jeden ze zde popsaných způsobů a máme vyhovující telefon (tím se rozumí téměř všechny mobilní telefony renomovaných značek), dostaneme tak mobilní telefon do stavu přizpůsobeného pro funkci *watchdog* realizovanou pomocí platformy Arduino, o které bude řeč v další části diplomové práce.

5.4.2 Konstrukce hlídacího obvodu pomocí platformy Arduino

O tom, k jakému účelu slouží hlídací obvod, jsem již psal. K jeho realizaci potřebujeme následující komponenty:

- **Vývojová deska Arduino Uno R3**

Mozkem a srdcem funkce *watchdog* je vývojová deska Arduino Uno R3, vyobrazená na obrázku 5.6. V paměti této desky je nahrán program, pomocí kterého je ovládán informační displej a relé. Veškeré informace zobrazené na informačním displeji jsou generovány pomocí již zmíněného programu. Taktéž je v programu Arduina nastaveno ovládání relé, pomocí kterého je simulováno vyjmutí a opětovné vložení baterie mobilního telefonu. Schéma celého zapojení i konektory Arduina, které jsou využity k napájení a ovládání relé a informačního displeje, jsou k vidění na obrázcích 5.12 a 5.13. Informace o tom, jak vytvářet programy a následně je nahrát do vývojové desky Arduino, se můžete dočíst v seznamu literatury [15].

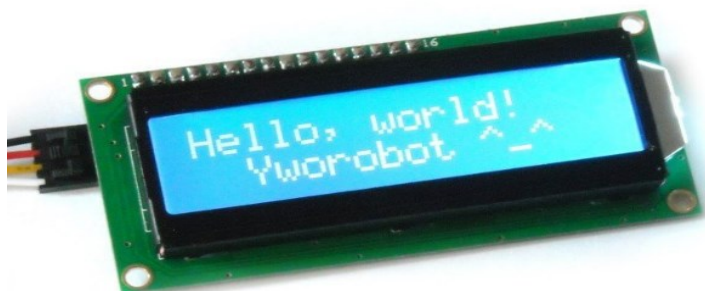


Obrázek 5.6: Vývojová deska Arduino Uno R3

- **Informační displej SPI LCD 1602**

Informační displej SPI LCD 1602, zobrazen na obrázku 5.7, může zobrazit až 16 znaků v každém ze dvou řádků. V testovací sondě je použit pouze k informačním účelům. Pomocí něj je uživatel informován o tom, v jakém stádiu se právě testovací sonda nachází a kolik času zbývá do dalšího preventivního restartování mobilního telefonu. Pro zpřístupnění používání displeje můžeme použít již existující knihovny *LiquidCrystal_I2C*, na niž je odkaz uveden

v seznamu použité literatury [16]. Zobrazení zbývajících času do restartu a dalších informací je pak poměrně jednoduché.



Obrázek 5.7: Informační displej SPI LCD 1602

- **Napájení vývojové desky Arduino**

Napájení Arduina je realizováno pomocí USB kabelu typu A-B. Ten je připojen k počítači nebo k nabíječce, která je připojena k síti. Druhý možný způsob napájení je pak pomocí bateriového boxu, který je k vidění na obrázku 5.8. Do bateriového boxu vložíme 6 baterií typu AA (tzv. tužkové baterie) o velikosti napětí 1,5 V. V případě použití baterií pak bude Arduino fungovat nezávisle na elektrické síti, nicméně doba, po kterou vystačí napájet desku Arduino, displej a relé bude omezeno jen na několik hodin či dní.



Obrázek 5.8: Bateriový box sloužící k napájení Arduina

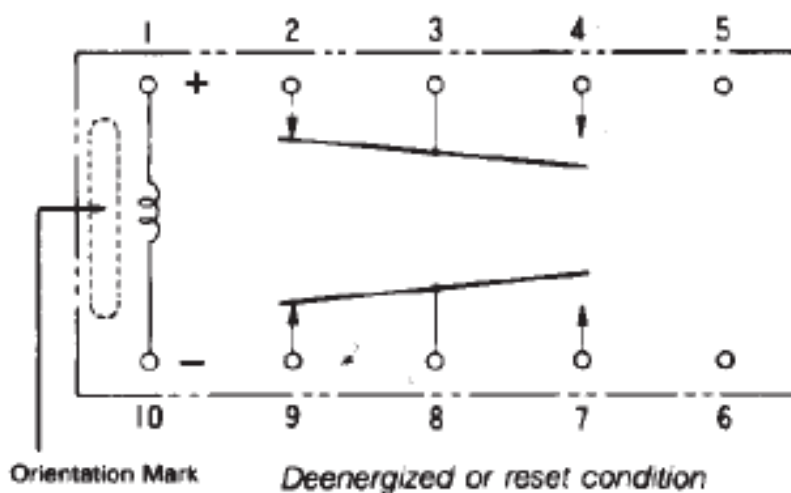
- **Relé s 2 přepínacími obvody**

Pro simulaci vyjmutí a opětovného vložení baterie zpět do mobilního telefonu je použito elektromagnetické relé se dvěma spínacími obvody. Dva spínací obvody kvůli nutnosti přerušit právě dvě spojení mezi baterií a mobilním telefonem. Kladný a záporný pól baterie. Prostřední konektory baterie, které jsou využívány například při nabíjení baterie, jsou připojeny k telefonu neustále pomocí vodičů. Není nutné je přerušovat při simulaci vytažení baterie.

V mém řešení jsem použil relé Fujitsu-Takamisawa A5W-K, zobrazeno na obrázku 5.9. To umožňuje spínat 2 obvody a je výhodné jej použít zejména kvůli jeho malé spotřebě, takže může být napájeno přímo pomocí Arduina. Katalogový list relé je k dispozici v seznamu literatury [17].



Obrázek 5.9: Relé Fujitsu-Takamisawa A5W-K

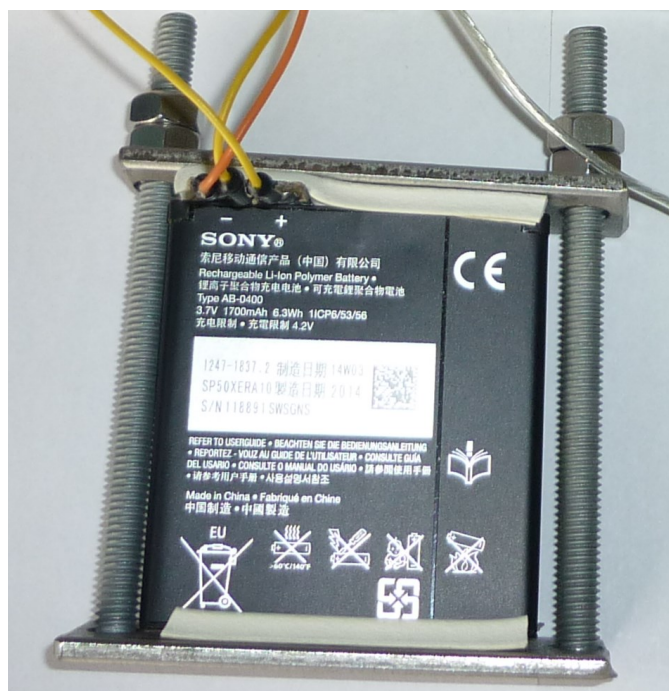


Obrázek 5.10: Schéma relé Fujitsu-Takamisawa A5W-K

Schéma relé Fujitsu-Takamisawa A5W-K je na obrázku 5.10. Ke spínání relé jsou použity konektory 1 a 10. Ke konektoru 1 je přivedeno napětí 5V z Arduina, k 10 je přivedena zem. Ke konektorům 3 a 8 je přivedeno plus a minus z baterie, vyvedení plusu a mínusu baterie je pak realizováno pomocí konektorů 4 a 7. Pokud přivedeme pomocí Arduina napětí na konektor 1, dojde k sepnutí přepínacích obvodů a je tím simulováno vložení baterie do mobilního telefonu. Když naopak napětí nepřivedeme, oba přepínací obvody jsou rozpojeny. Dojde k vyjmutí baterie.

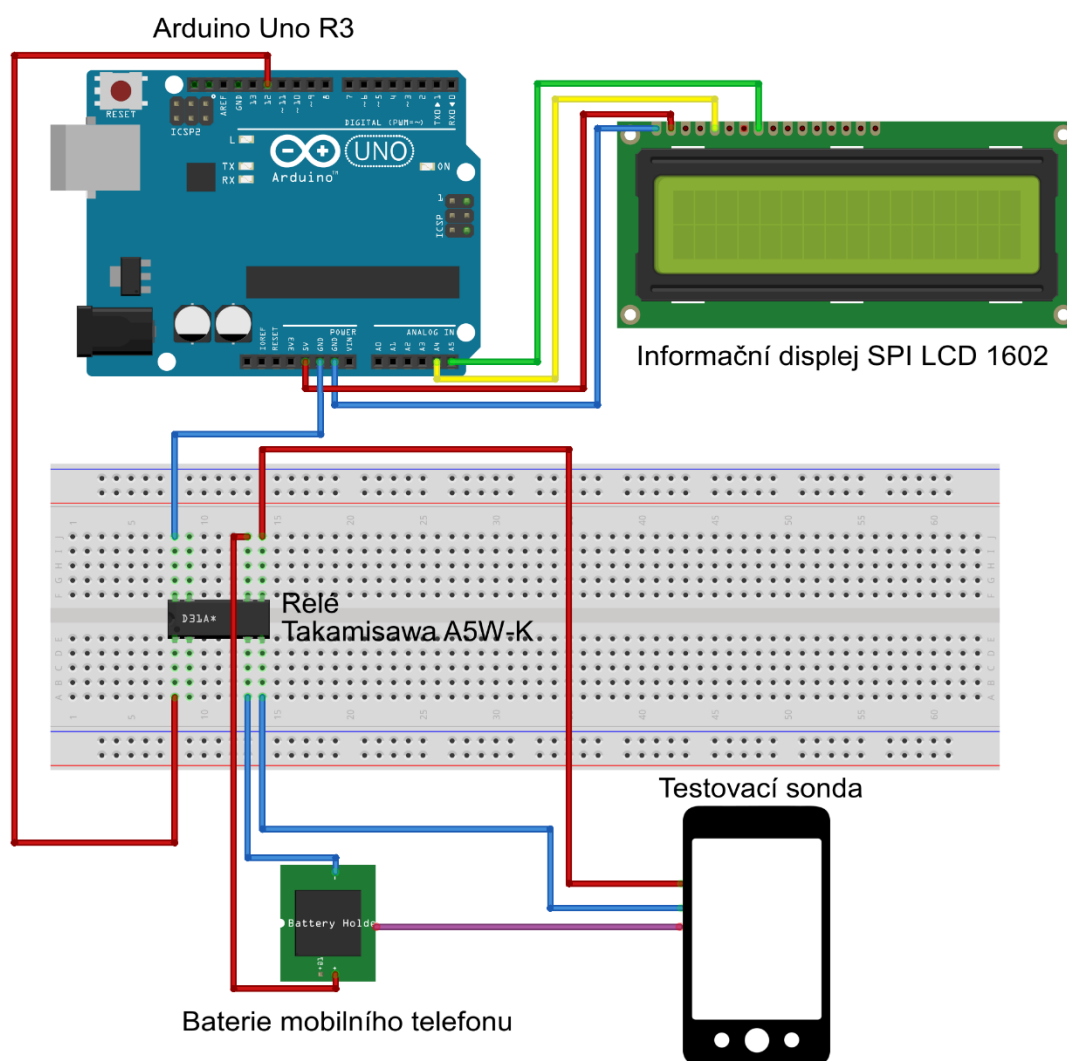
- **Externí držák baterie, vodiče a konektory**

K propojení veškerých komponent jsou použity klasické jednožilové vodiče. Díky konstrukci vývojové desky Arduino i použitého informačního displeje SPI LCD 1602 nemusíme vůbec pájet, vodiče stačí pouze zastrčit do připravených portů. Jediné, kde se pájí, tak je v místě napojení vodičů na mobilní telefon a jeho baterii. Nicméně i to lze obejít použitím například plošných spojů, na které napájíme vodiče. Tyto plošné spoje pak mechanicky přitlačíme na příslušné konektory baterie a mobilního telefonu a vyhneme se tak pájení vodičů přímo na baterii či telefon, které by mohlo tyto součásti nevratně poškodit. K realizaci držáku baterie, který má za úkol držet baterii přitisknutou k plošnému spoji, jsem použil 2 dlouhé šrouby, 4 matky a 2 obdélníkové kovové plošky. Tyto plošky jsou pomocí šroubů a matek utahovány k sobě a zajišťují tak kontakt mezi baterií a plošným spojem, na kterém jsou napájeny příslušné vodiče. Realizace držáku baterie je na obrázku 5.11. Podobný princip je pak použit i v případě spojení mezi plošným spojem a konektory mobilního telefonu, na kterých je očekáváno napájení pomocí baterie.

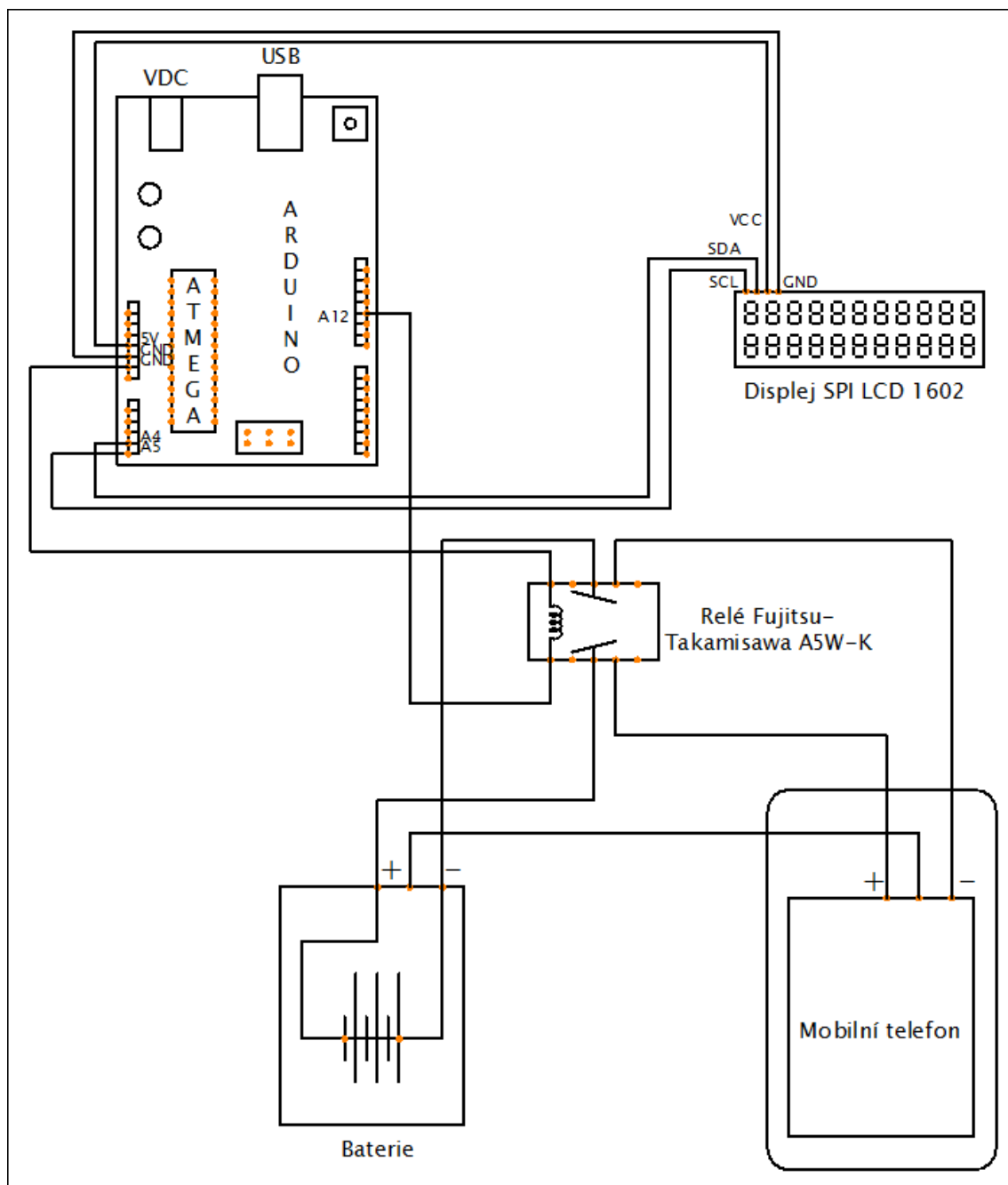


Obrázek 5.11: *Držák baterie s připojenými konektory*

Realizace kompletní funkce *watchdog* je k vidění na názorných schématech na obrázcích 5.12 a 5.13. Je zde vyobrazena vývojová deska Arduino Uno R3, informační displej SPI LCD 1602, relé pro spínání dvou obvodů Fujitsu Takamisawa A5W-K vložené do nepájivého pole, externě umístěná baterie vložená v držáku mimo mobilní telefon a testovací sonda v podobě mobilního telefonu.



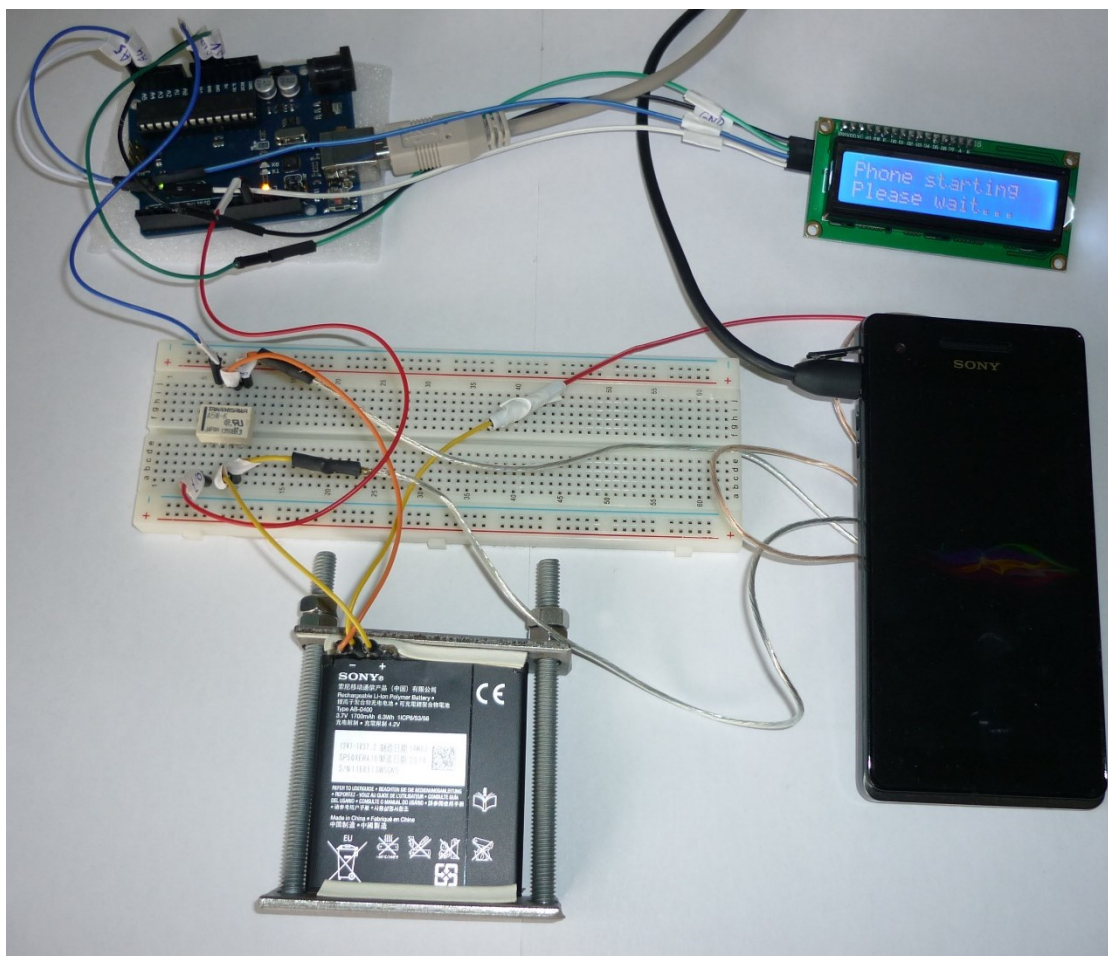
Obrázek 5.12: Schéma funkce watchdog vytvořená v programu Fritzing



Obrázek 5.13: *Elektrické schéma řešení nakreslené pomocí programu ProfiCAD*

Schéma řešení z obrázků 5.12 a 5.13 je pak vyfoceno i realizované v praxi na obrázku 5.14. Všechny části jsou rozloženy od sebe, aby bylo viditelné složení celého řešení i propojení jeho jednotlivých částí. Toto řešení je již plně funkční a použitelné v praxi. Nicméně vylepšení

hlavně z pohledu možného přenášení testovací sondy by bylo v tom směru, že celá testovací sonda včetně hlídacího obvodu by byla zapouzdřená do plastové krabice odpovídajících rozměrů. Mám vyzkoušeno, že se žádná z použitých komponent i během dlouhodobějšího provozu nijak výrazněji nezahřívá, proto zapouzdření do plastové krabice by nemělo být problémem. Zjednodušilo by se tak možné přenášení sondy mezi různými lokacemi.



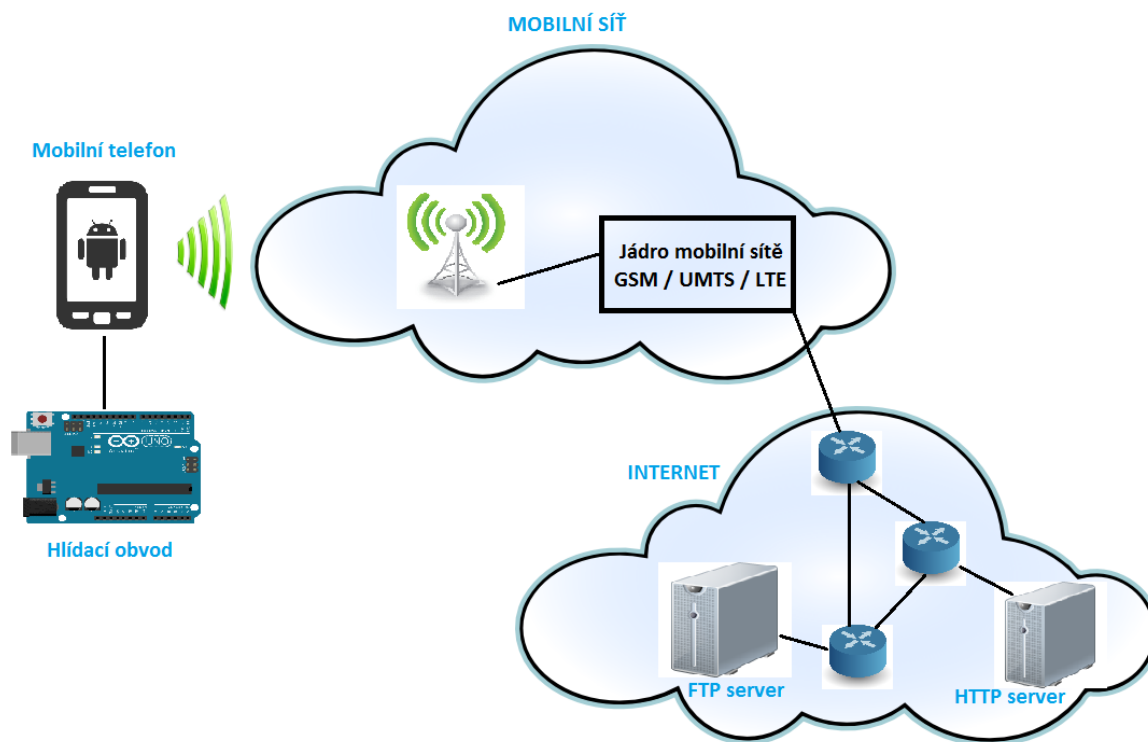
Obrázek 5.14: Realizace testovací sondy

6 Testování sondy

V následující kapitole bude popsán postup, jakým byla testována sonda v reálných podmínkách mobilní sítě GSM, UMTS a LTE.

6.1 Schéma testování

Na obrázku 6.1 je zobrazeno, jakým způsobem probíhá testování. Mobilní telefon Sony Xperia LT25i, jehož napájení, potažmo restartování je řízeno pomocí hlídacího obvodu složeného z Arduina a dalších komponent, přistupuje skrze mobilní datovou síť T-Mobile do Internetu. Ze vzdáleného HTTP serveru stahuje datový soubor a poté na jiný vzdálený FTP server soubor posílá. Takto provede stanovení přenosových rychlostí a dalších parametrů pro všechny dostupné typy mobilních datových sítí.



Obrázek 6.1: Schéma testování

6.2 Zjištěné údaje

Aby testování odpovídalo specifikaci měření dle doporučení ETSI TR 102 678, musíme zajistit, aby se přenášený soubor nestihl během předem stanovené doby měření přenést celý. Nejdříve je tedy nutné nastavit dostatečnou dobu měření, aby výsledky byly co nejpřesnější. Na druhou stranu nesmí být doba přenosu přehnaně dlouhá, abychom zbytečně nezatěžovali síť. Jako dostatečně dlouhá doba pro měření rychlostí stahování a odesílání dat byla zvolena doba 10 sekund. Při měření po dobu kratšího časového intervalu byly výsledky dosti nepřesné, měření nebylo spolehlivé. Pokud jsem naopak nastavil větší interval, výsledky byly velmi podobné jako při měření po dobu 10 sekund. Čím delší dobu je měření prováděno, tím je větší zatížení sítě. Proto bylo zvoleno jako optimální doba měření právě interval 10 sekund, který nabízí získání poměrně přesných a spolehlivých výsledků při zachování co nejmenšího zatížení sítě.

Druhý parametr, který se podařilo zjistit až během testování v reálném provozu, je velikost stahovaného a odesílaného souboru pro měření rychlostí. Velikost musí být větší, než je možné přenést při teoreticky největších možných přenosových rychlostech mobilních sítí během 10 sekund. Zvolená velikost souboru pro stahování je 100 MB a soubor pro odesílání má velikost 50 MB. Obě zvolené velikosti jsou dostatečně velké.

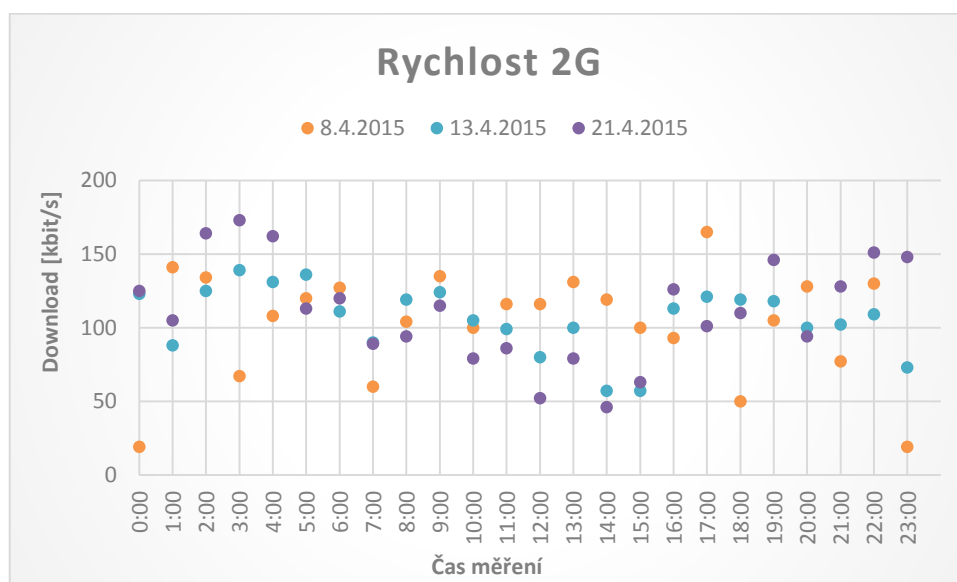
Další věcí, která se projevila až během testování aplikace v praxi je fakt, že bylo nutné nastavit ochrannou dobu mezi přepnutím se do daného typu mobilní sítě a následným otestováním této mobilní sítě. Ochranný interval byl nastaven na 60 sekund a poskytuje tak mobilnímu telefonu dostatečný čas pro přepnutí se do požadované mobilní sítě. Taktéž bylo nutné nastavit ochranný limit po zapnutí mobilního telefonu. Limit pro započítání testování je nastaven na 3 minuty od zapnutí mobilního telefonu.

Aby byla co nejvíce vypovídající hodnota odezvy, je příkaz *ping* proveden 20 krát a výsledná odezva je pak průměrem všech dvaceti provedených pokusů.

7 Naměřené výsledky

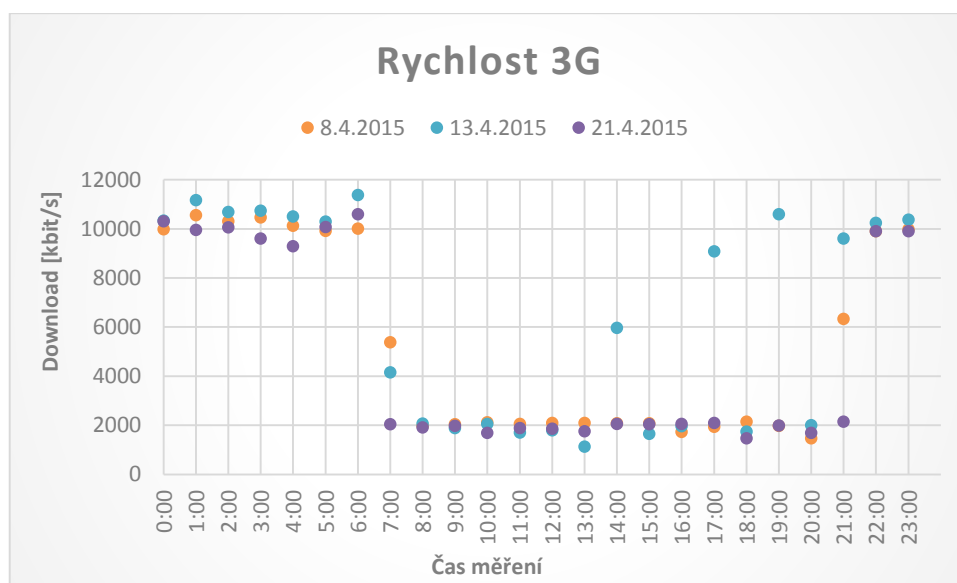
V této kapitole zobrazím výsledky měření získané pomocí testovací sondy. Veškerá měření zde zobrazená probíhala pomocí testovací sondy Sony Xperia V. Tento telefon byl umístěn uvnitř rodinného domu v obci Chlebičov. Ten se nachází ve vzdálenosti 700 metrů vzdušnou čarou od základnové stanice T-Mobile, na kterou byl mobilní telefon během měření připojen. Je nutné zdůraznit, že se jedná pouze o experimentální měření, jejichž cílem je ověřit funkčnost testovací sondy. Konkrétní měření, jejich výsledky ani statistiky nejsou předmětem této diplomové práce.

7.1 Výsledky měření rychlosti stahování dat sítě 2G



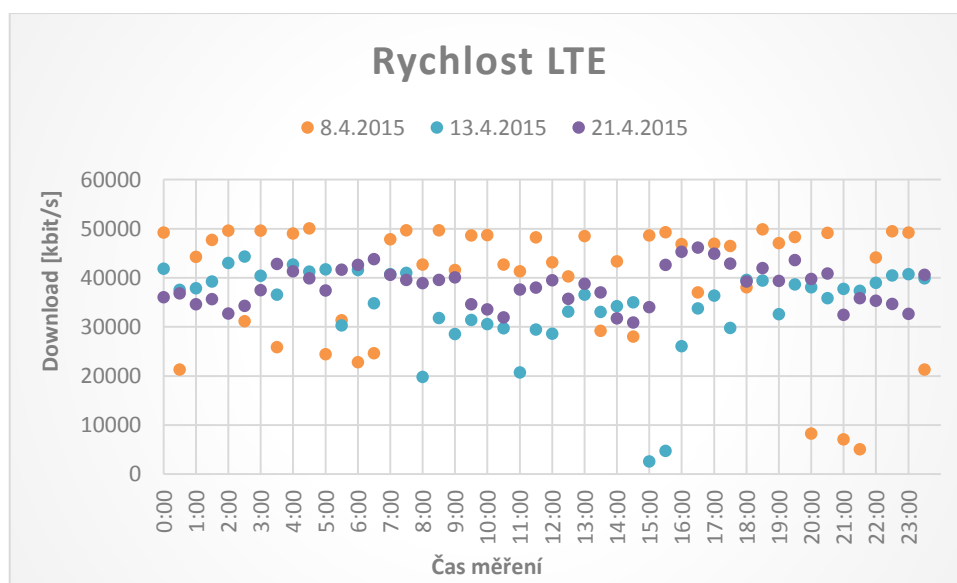
Obrázek 7.1: Výsledky testování rychlosti sítě GSM

7.2 Výsledky měření rychlosti stahování dat sítě 3G



Obrázek 7.2: Výsledky testování rychlosti sítě 3G

7.3 Výsledky měření rychlosti stahování dat sítě LTE



Obrázek 7.3: Výsledky testování rychlosti sítě LTE

Vyhodnocení naměřených hodnot není předmětem této diplomové práce. Obecně však lze říci, že vyšších rychlostí je dosahováno v nočních hodinách, zatímco přes den, v době

špičky, jsou naměřené hodnoty rychlosti stahování spíše nižší. Grafy na obrázcích 7.1, 7.2 a 7.3 tak demonstrují reálné fungování testovací sondy, jejich význam však zůstává pouze informační.

Závěr

Tato diplomová práce je zaměřená na monitoring přenosových parametrů mobilních sítí. Především ve své první části, která je spíše teoretická, se zabývá mobilní sítí. Je zde popsána historie mobilních sítí od prvního bezdrátového přenosu informace v roce 1865 až po současnost. Dále jsou zde vymezeny mobilní sítě dle způsobu šíření signálu a rozdělení sítí podle historického vývoje od nejstarších sítí spadajících do nulté generace až po nejnovější síť čtvrté generace, které se v současné době hojně rozšiřují i na území České republiky. Důraz je kladen především na síť GSM, UMTS a LTE, které jsou tématem této práce.

Dále rozebírám pojem kvalita služby pro mobilní technologie, jakými již existujícími a především drahými laboratorními přístroji lze kvalitu mobilních sítí stanovit a jaké parametry slouží k určení právě této kvality služby.

Třetí kapitolu tvoří informace z reálné praxe. Zabývá se testováním a optimalizací mobilní sítě tak, jak probíhá v reálných podmínkách v síti T-Mobile Czech Republic a.s. Rozděluje tyto procesy do třech dílčích podskupin, kdy každá z podskupin má specifický účel využití získaných měření a využívá k němu různé přístroje, které jsou zde vyjmenovány. Dovídáme se, že celkově se touto problematikou u mobilního operátora T-Mobile v České republice zabývá asi 20 lidí, a že získané výsledky jsou využity k optimalizaci mobilní sítě, snížení nákladů a zkvalitnění poskytovaných služeb zákazníkům.

Čtvrtou část tvoří informace o použitých technologiích v rámci řešení diplomové práce, konkrétně se jedná o tři oblasti. Operační systém mobilních telefonů Android, na který je vývoj testovací sondy zaměřen, dále platforma Arduino, pomocí které je realizována funkce *watchdog* a mobilní telefon Sony Xperia V, na kterém probíhal kompletní vývoj testovací aplikace a představuje tak vyvinutou testovací sondu.

Hlavní náplní diplomové práce je část, která se zabývá přímo vývojem testovací sondy. Je zde zmíněno, proč byla dána přednost realizace testování pomocí skriptů napsaných v *Bashi* před vývojem programu v některém z klasických programovacích jazyků. Testování je implementováno pomocí tří skriptů. Inicializační skript, který automaticky po zapnutí mobilního telefonu inicializuje testování. Dále hlavní měřicí skript, který provádí veškerá měření, zjišťování důležitých parametrů a odeslání výsledků v požadované podobě na vzdálený FTP server. Doplnkový skript pak pomocí AT příkazů získává hodnoty LAC a Cell ID, které jsou součástí výsledného CSV souboru s logem obsahujícím zjištěné informace během testování mobilní sítě. Je zde zobrazen postup měření, jakým způsobem je stanovena rychlost tak, aby odpovídala specifikaci měření uvedené v technické zprávě ETSI TR 102 678. Dozvíme se zde, jakým způsobem lze programově přepnout typ mobilní sítě, jak pomocí AT příkazů zjistit parametry LAC a Cell ID, jak pomocí nástroje *ping* stanovit délku zpoždění, a také o využití kombinace nástrojů *dumppsys*, *getprop* a informací obsažených ve virtuální adresářové struktuře */proc/* pro zjištění dalších parametrů.

Další praktická část se zabývá vývojem a implementací funkce *watchdog*. Ta je realizována pomocí platformy Arduino a sestává z těchto částí: vývojová deska Arduino Uno R3, informační displej SPI LCD 1602, relé pro spínání dvou nezávislých obvodů Fujitsu Takamisawa A5W-K, externí držák baterie a s ním související maketa baterie vložená do mobilního telefonu a dále jednotlivá propojení pomocí konektorů a jednožilových vodičů. Je zde popsáno nutné přizpůsobení mobilního telefonu s operačním systémem Android tak, aby se v době nabíjení při vyndání baterie vypnul a při opětovném vložení baterie do zařízení aby se automaticky a samočinně opět zapnul.

V další kapitole je znázorněno, jakým způsobem probíhalo testování vyvinuté sondy v reálných podmínkách v síti T-Mobile. Díky předešlému testování jsem mohl správně nastavit jednotlivé ochranné časové intervaly mezi jednotlivými událostmi, nastavit správnou velikost souborů, které se přenášejí pro stanovení přenosových rychlostí, a také stanovit dostatečný časový interval stahování a odesílání dat pro výpočet přenosových rychlostí. Tento časový interval je dostatečně dlouhý, aby poskytoval korektní výsledky a zároveň nadbytečně nezatěžoval mobilní datovou síť dlouhým měřením.

Poslední kapitola je pak věnována konkrétním výsledkům, které se mi podařily pomocí vyvinuté testovací sondy získat. Veškeré zde zobrazené výsledky pocházejí z měření v mém bydlišti – obci Chlebičov u Opavy. Nicméně v době testování sondy jsem prováděl měření také v Ostravě v areálu VŠB a v Brně v budově T-Mobile, kde byly mnou získané výsledky porovnávány s výsledky získanými pomocí komerčních sond. Zjistili jsme, že výsledky získané pomocí mé testovací sondy jsou velmi podobné výsledkům získaným pomocí komerčního nástroje.

Co se týká možného dalšího vývoje produktu, jedná se především o možnost vytvoření hlasového hovoru automaticky pomocí kódu. Tento hovor, který by byl veden na speciální číslo, by mohl být analyzován a mohly by být takto získány komplexní výsledky týkající se datových přenosů i hlasových hovorů. Nicméně tato oblast už nebyla předmětem diplomové práce.

Použitá literatura

- [1] HEINE, Gunnar. *GSM Networks: Protocols, Terminology and Implementation*. 1. vyd.: Artech House, 1999, 416 s. ISBN 0890064717.
- [2] MUHAMMAD, Faris. *An Introduction to UMTS Technology*. 1. vyd.: BrownWalker Press, 2008, 311 s., ISBN 1-1599-446-0.
- [3] RUMNEY, Moray. *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges, 2nd Edition*. 2. vyd.: Agilent Technologies, 213, 648 s., ISBN 978-1-119-96257-1.
- [4] Pokrytí | Veřejné širokopásmové mobilní sítě. In: Český telekomunikační úřad [online]. 2014 [cit. 2015-04-22]. Dostupné z: <http://lte.ctu.cz/pokryti/>
- [5] TYLER, Jason. *Android Hacker's Toolkit*. 1. vyd.: Courier Westford, 2012, 192 s., ISBN 978-1-119-95138-4.
- [6] Arduino - Home. In: Arduino [online]. 2015 [cit. 2015-04-22]. Dostupné z: <http://www.arduino.cc/>
- [7] Keeping the Device Awake. In: Android Developers [online]. 2015 [cit. 2015-04-22]. Dostupné z: <https://developer.android.com/training/scheduling/wakelock.html>
- [8] Settings – simple way to toggle between connections. In: Android Stack Exchange [online]. 2015 [cit. 2015-04-22]. Dostupné z: <http://android.stackexchange.com/questions/44347/simple-way-to-toggle-between-2g-and-3g-connection>
- [9] Just do IT. In: Android Property System [online]. 2010 [cit. 2015-04-22]. Dostupné z: <http://rxwen.blogspot.cz/2010/01/android-property-system.html>
- [10] GNU Wget. In: The GNU Operating System [online]. 2012 [cit. 2015-04-01]. Dostupné z: <https://www.gnu.org/software/wget/>
- [11] ETSI.org. In: ETSI - European Telecommunications Standards Institute [online]. 2011 [cit. 2015-04-07]. Dostupné z: http://www.etsi.org/deliver/etsi_tr/102600_102699/102678/01.02.01_60/tr_102678v010201p.pdf
- [12] SignalStrength. In: Android Developers [online]. 2015 [cit. 2015-04-22]. Dostupné z: <http://developer.android.com/reference/android/telephony/SignalStrength.html>
- [13] Android Radio Layer Interface. In: Slideshare [online]. 2015 [cit. 2015-04-22]. Dostupné z: <http://www.slideshare.net/ssusere3af56/android-radio-layer-interface>

-
- [14] AT+CREG – Network. In: M2MSupport.net [online]. 2015 [cit. 2015-04-22]. Dostupné z: <http://m2msupport.net/m2msupport/atcreg-network-registration/>
- [15] MARGOLIS, Michael. *Arduino Cookbook*. 2. vyd.: Sebastopol: O'Reilly, 2012, 699 s., ISBN 978-1-449-31387-6.
- [16] LCD-Blue-I2C. In: Arduino-info [online]. 2015 [cit. 2015-04-22]. Dostupné z: <https://arduino-info.wikispaces.com/LCD-Blue-I2C>
- [17] A5W-K datasheet. In: Datasheet4U [online]. 2015 [cit. 2015-04-22]. Dostupné z: http://www.datasheet4u.com/datasheet/A/5/W/A5W-K_FujitsuMicroelectronics.pdf.htm